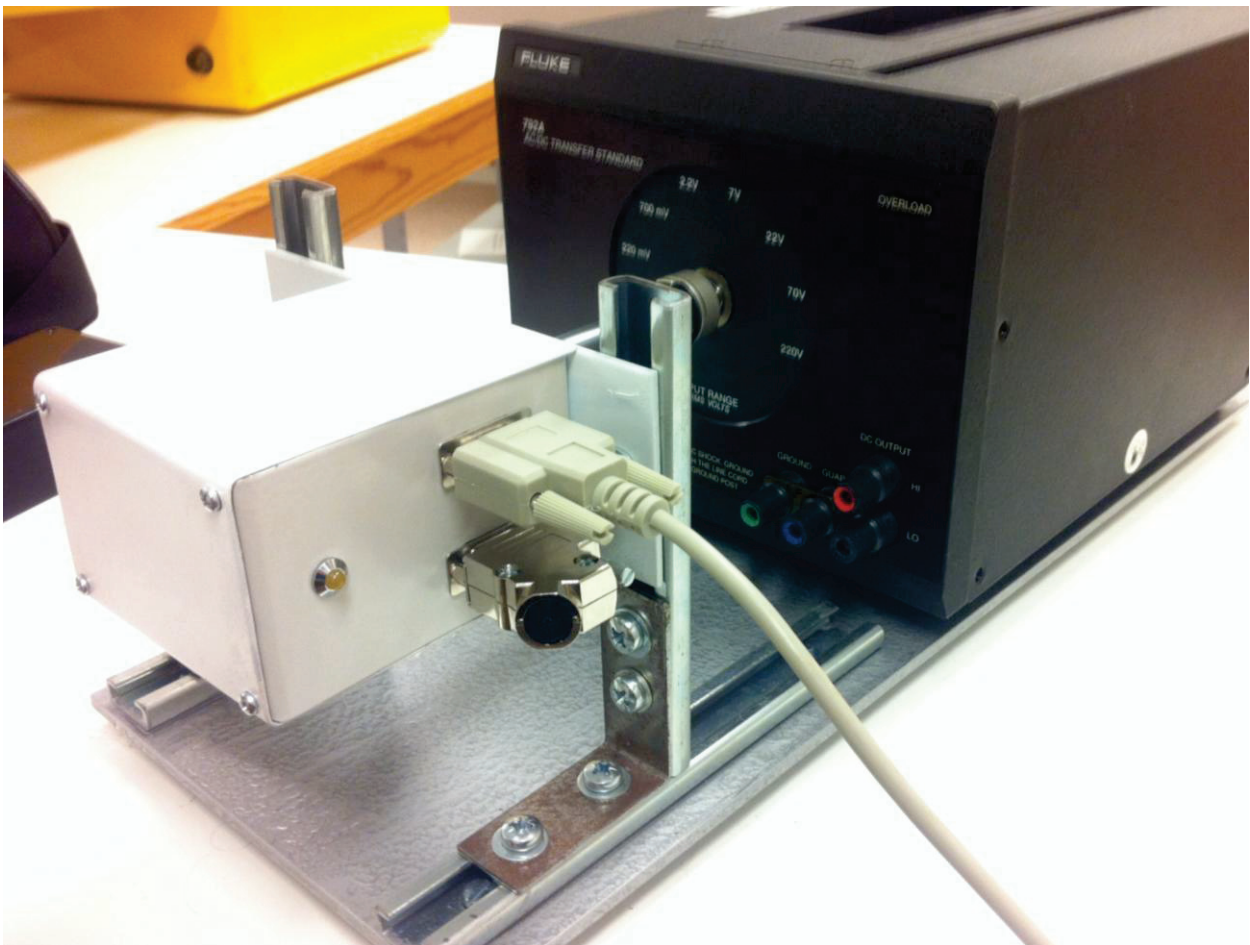




CHALMERS



Automatisk motorstyrning av vridomkopplare

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

ANTON LARSSON
MARCUS DAVIDSON

Automatisk motorstyrning av vridomkopplare

Examensarbete inom
högskoleingenjörsprogrammet Mekatronik

Marcus Davidson

Anton Larsson

2014-05-21

FÖRORD

Detta är ett examensarbete utfört av Marcus Davidson och Anton Larsson på institutionen för signaler och system på Chalmers tekniska högskola. Arbetet utfördes på Sveriges tekniska forskningsinstitut (SP) i Borås. Det utgjorde avslutningen på utbildningen till mekatronikingenjör och innefattar 15 högskolepoäng av programmets totala 180 högskolepoäng.

Vi vill tacka Peter Erlandsson som var vår första kontakt på SP. Dessutom vill vi tacka våra handledare – Karl-Erik Rydler och Valter Tarasso på SP som har varit med under arbetets gång och Manne Stenberg på Chalmers som hjälpt oss med rapportskrivandet. Vi är också tacksamma för resten av personalen på elektrisk mätteknik på SP som har sett till att vi fått tillgång till allt vi behövde och bidragit med sin kunskap och erfarenhet!

SAMMANFATTNING

SP i Borås behövde en nyutveckling av en automatisk omkopplingsvridare, då deras gamla system slitits ut. Utrustningen skulle användas för att möjliggöra att mätningar kunde pågå dygnet runt och inte vara beroende av en närvarande tekniker som ställer om mätinstrumenten mellan olika spänningsområden. Under 10 veckors tid utfördes ett examensarbete av två studenter från Chalmers tekniska högskola med målet att utveckla denna utrustning. Systemet som togs fram är baserat på en stegmotor med en beröringsfri lägesgivare för att uppnå lång livslängd och ett inbyggt system som mäter motorns vridmoment. Det kan både självkalibrera sig emot de omkopplare som det ansluts till och konfigureras manuellt, och det avger inga mätbara induktiva eller kapacitiva störningar som kan manipulera de känsliga mätningar som inkopplade instrument utför. Det finns även möjlighet att seriekoppla enheter för styrning av flera omkopplare samtidigt. Styrning av utrustningen sker genom program utvecklade i LabVIEW, som även styr mätutrustningen och samlar in data. Som ett tillägg till uppgiften utvecklades även ett LabVIEW-program för styrning av en batterihanterare för det huvudsakliga mätinstrumentet *Fluke 792A*.

ABSTRACT

The Technical Research Institute of Sweden (SP) was in need of a redevelopment of an automatic rotary switch controller, since their previously used system was no longer working. The equipment would allow measuring experiments to be conducted 24 hours a day and not be dependant of a present engineer to adjust settings. A degree project was conducted during 10 weeks by two students from Chalmers University of Technology to develop the new system. The finished product is based on a stepper motor with a load measuring system and a contactless rotary encoder to achieve a long lifespan. It is able to self-calibrate and automatically find the steps in the connected rotary switch, but it also has a manual way of configuring steps. It is tested to not emit any inductive or capacitive disturbances that can interfere with measurements made by connected instruments. Multiple units can be connected to, and operated by, the same computer at the same time. The equipment is controlled by software developed in LabVIEW. As an added objective, a LabVIEW-program was also developed to communicate with a battery pack controller for Fluke 792A, which was the primary measuring instrument for the equipment to be used with.

INNEHÅLL

1	INLEDNING.....	1
1.1	BAKGRUND	1
1.2	SYFTE.....	1
1.3	AVGRÄNSNINGAR.....	1
1.4	PRECISERING AV UPPGIFTEN.....	1
2	METOD.....	2
3	TIDSPLAN.....	3
4	TEKNISK BAKGRUND	4
4.1	LABVIEW.....	4
4.1.1	Användningsområden.....	4
4.1.2	Utvecklingsmiljö.....	4
4.2	ELMOTORER	5
4.2.1	Likströmsmotorn	5
4.2.2	Stegmotorer	6
5	UTVECKLING AV HÅRDVARA.....	8
5.1	VAL AV MOTORTYP	8
5.2	UPPBYGGNAD AV SYSTEMET	8
5.2.1	Centralenheten	8
5.2.2	Motorenheterna.....	9
5.2.3	Störningsmätning vid motorenheter	9
5.3	MONTERINGSRIGG FÖR FLUKE 792A.....	11
6	UTVECKLING AV PROGRAMVARAN.....	12
6.1	KOMMUNIKATION	12
6.2	KALIBRERING.....	13
6.2.1	Vridmoment.....	13
6.2.2	Stegkalibrering.....	14
6.3	POSITIONERINGSPROGRAM	18
6.4	ENGÅNGSINITIERING AV PANDRIVE-ENHETER.....	18
6.5	ANPASSNING EFTER STYRANDE MJUKVARA.....	19
7	ANVÄNDNING AV SYSTEMET.....	20
7.1	BRA ATT VETA OM SYSTEMET.....	20
7.1.1	Ihopkoppling av hårdvara	20
7.1.2	Kalibrering efter omkopplare.....	20
7.1.3	Styrning av omkopplare	21
7.2	MJUKVARANS KOMPONENTER.....	21
7.3	KALIBRERING AV MODULER	22
7.3.1	Ladda konfiguration från fil.....	23
7.3.2	Kalibrera nytt instrument	24
7.4	POSITIONSANROP.....	30
7.4.1	Fristående version.....	30
7.4.2	VI-version.....	31
7.5	INSTALLATION AV MOTORENHETER	32
8	RESULTAT OCH DISKUSSION.....	33
8.1	UPPFÖLJNING AV MÅLSÄTTNING	33
8.2	ARBETSGÅNG	34

8.3	ÅTERKOPPLING TILL TIDSPLANEN	34
8.4	ALLMÄN DISKUSSION	35
9	SLUTSATS	35
	REFERENSLISTA	36
	BILAGOR.....	37

Förkortningar:

SP	Sveriges tekniska forskningsinstitut
CE	Conformité Européenne (märkning som visar att en produkt är godkänd utifrån EU-direktiv)
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
VI	Virtual instrument (utvecklas i LabVIEW)
RS-232	Recommended Standard 232 (seriell datakommunikation)
RS-485	Recommended Standard 485 (seriell datakommunikation)
ms	Millisekund
D-sub	D-formad kontaktstandard
COM-port	Seriell kommunikationsport (från början ett fysiskt gränssnitt, men i detta projekt används virtuella COM-portar)

1 INLEDNING

En programmerbar automatisk omkopplingsvridare ska konstrueras på SP – Sveriges tekniska forskningsinstitut.

1.1 Bakgrund

På elektrisk mätteknik på SP utförs långa mätningsserier där det krävs att mätinstrument ställs om mellan olika elektriska spänningsområden mellan testerna. Detta gör att någon ständigt måste passa upp dessa mätningar och ställa om utrustningen för att undvika att den bara står och väntar. Ett examensarbete utfördes för ungefär 20 år sedan för att effektivisera denna process och spara personaltid och möjliggöra att mätningar pågår under dygnets alla timmar. En automatisk omställare togs då fram som innan varje mätning ställde in mätutrustningen till önskat läge. Den bestod av en motor och en växellåda som vred på mätinstrumentets omkopplare. Alla reservdelar har nu slitits ut och finns inte längre att få tag på, så en nyutveckling krävs.

1.2 Syfte

Syftet med projektet är att konstruera en anordning som automatiskt vrider omkopplaren på en mätutrustning till rätt läge för den mätning som ska utföras. Anordningen ska kunna kalibrera sig efter det instrument den kopplas till, det vill säga hitta de olika lägen som omkopplaren kan vridas till. Anordningen ska uppfylla aktuella direktiv från EU (exempelvis lågspänningsdirektiv och direktiv för elektromagnetisk kompatibilitet) och ska vara konstruerad så att den inte ger störningar hos mätutrustningens värden. Tanken är att den färdiga produkten ska provas för CE-märkning.

1.3 Avgränsningar

Testning för CE-märkning kommer inte genomföras under projektiden.

De mätningar som anordningen ska användas till att utföra kommer inte behandlas i projektet, då de redan sedan länge är i drift.

1.4 Precisering av uppgiften

En produkt ska tas fram för automatisk styrning av roterande omkopplare, det främsta syftet är för att automatisera mätinstrumentet *Fluke 792A*. Produkten ska uppfylla följande:

- Den ska vara enkel att montera mot en omkopplare
- Det ska gå att köra minst två enheter parallellt ifall man har ett instrument där flera omkopplingsmöjligheter finns.
- Enheten ska inte ge ifrån sig störningar som kan påverka känsliga mätningar.
- Enheten ska kunna kalibrera sig själv efter aktuell omkopplare.
- Livslängden ska vara längre än föregående utrustning.

2 METOD

Anordningen ska tas fram från grunden, det vill säga både mekaniken, elektroniken och mjukvaran ska konstrueras. Den ursprungliga anordningen ska studeras för att se vad man vill behålla och vad som ska ändras. Till det användes en likspänningsmotor som matades med konstant spänning vid körning. Strömmen till motorn mättes upp för att beräkna dess belastning och därmed hitta omkopplarens olika lägen. Som lägesgivare användes en potentiometer som satt på motoraxeln. När systemet nu ska nykonstrueras ska det undersökas om detta fortfarande är det bästa alternativet, eller om andra alternativ är mer aktuella idag, 20 år senare. Exempelvis skulle en stegmotor kanske kunna användas, samt någon annan typ av lägesgivare.

Arbetet kommer utföras på plats, det vill säga på SP i Borås. Mjukvaran ska utvecklas i LabView och till det finns stora möjligheter till hjälp och handledning inom SP. Kravspecifikationer ska tas fram för de olika delarna i systemet, kraven på elektroniken kommer att styras av EU-direktiv.

3 TIDSPLAN

Kortfattat ska projektet genomföras enligt följande förlopp:

1. Kravspecifikationer fastställs för huvudkomponenterna:
 - Den mekaniska konstruktionen med chassi, motor och växellåda.
 - Den elektroniska konstruktionen med strömförsörjning, givare och kommunikationsgränssnitt mot en styrande dator.
 - Programvaran som ska styra anordningen.
2. Utifrån kravspecifikationen ska lämplig hårdvara väljas. Detta innefattar bland annat att välja en motor med utväxling som ger tillräckligt vridmoment och precision.
3. Hela konstruktionen med mekanik och elektronik ska monteras ihop.
4. Mjukvara ska utvecklas för att styra anordningen.

Dokumentation och rapportskrivning sker löpande genom projektet. Nedan följer ett gantt-schema som beskriver den beräknade tidsplanen.

		Tids- och ansvarsplan										
Aktivitet / Vecka	Huvudansvariga	12	13	14	15	16	17	18	19	20	21	22
1. Projektadministration												
1.1 Kravspecifikation	Marcus & Anton											
1.1.1 Mekanik												
1.1.2 Elektronik												
1.1.3 Programmering												
1.1.4 Sammans komplett kravspecifikation												
1.2 Rapportskrivning	Marcus & Anton											
2. Projekt - konstruktion												
2.1 Mekanik	Marcus											
2.1.1 Fästänordning												
2.1.2 Dimensionering av motor & växel												
2.1.3 Montering												
2.2 Elektronik	Marcus & Anton											
2.2.1 Strömförsörjning												
2.2.2 Kommunikationsgränssnitt												
2.2.3 Lägesgivare												
2.2.4 Strömmätning												
2.3 Mjukvara	Anton											
2.3.1 Kommunikationsgränssnitt												
2.3.2 Självkalibrering												
2.3.3 Positionering												

3.1: Tidsplanering i Gantt-schema

4 TEKNISK BAKGRUND

Nedan följer en beskrivning av de tekniker och verktyg som användes under projektets gång.

4.1 LabVIEW

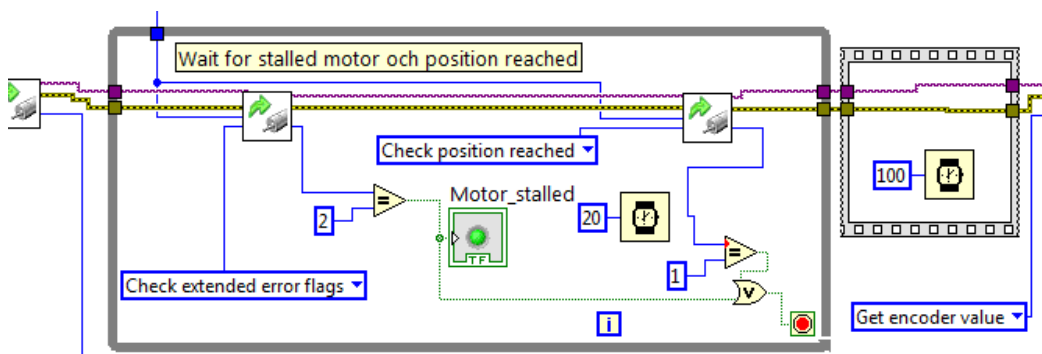
I detta avsnitt beskrivs kortfattat hur programmet LabVIEW är uppbyggt och används.

4.1.1 Användningsområden

LabVIEW är ett mycket flexibelt program men det största användningsområde är att kommunicera och styra hårdvara, genom en mängd olika plug-in-program kan nästan vilket gränssnitt som helst användas så länge det går att koppla till en dator. Det gör till exempel LabVIEW till ett utmärkt program för testning av hårdvara men även för att styra och samla in information ifrån olika enheter.

4.1.2 Utvecklingsmiljö

LabVIEW är ett program som bygger på ett grafiskt gränssnitt. Ett program byggs upp av flera småprogram likt subrutiner som kallas Virtuella Instrument som förkortas VI i LabVIEW. Varje VI har ett panelfönster som visar önskad information och ett blockdiagramsfönster där logiken byggs upp i form av olika funktionsblock och interna VI som kopplas samman genom ledningar. När ett program körs så sker allt parallellt i så stor utsträckning som möjligt så fort ett block har fått in alla signaler som den vill ha så körs blocket. För att undvika att saker körs i en oönskad ordning finns sekvensramar för att utföra saker i en bestämt följd. Nedan följer en exempelbild på gränssnittet. I bilden visas en del av programmet som utvecklats i projektet, där ett VI skrivet för att kommunicera med motorn (se avsnitt 6.2) anropas i en loop som inväntar en bestämd position.



Figur 4.1: Exempelbild på den grafiska programmeringsmiljön i LabVIEW.

Varje VI är som sagt ett eget lite program, det kan ta in och skicka ut information genom att välja vad som ska vara ingång respektive utgång. Panelfönstret är själva körfönstret som visas när programmet körs, det är önskad information skall visas och val och redigeringsmöjligheter ska finnas i form av knappar, reglage och textfönster. Körs ett VI internt i ett annat VI så går det att välja om panelfönstret skall visas eller inte.

En vanlig uppbyggnad av ett program är ett huvud-VI som visar den mest aktuella informationen och reglagen för att kunna utföra den tänkta uppgiften. Alla interna funktioner och inställningar får egna VI i så stor utsträckning som möjligt för att ge en bra överblick ifall något skall ändras eller felsökas. Finns det många inställningar kan det vara lämpligt att ha ett

inställnings-VI med ett eget fönster som visar alla inställningar samlade på ett och samma ställe.

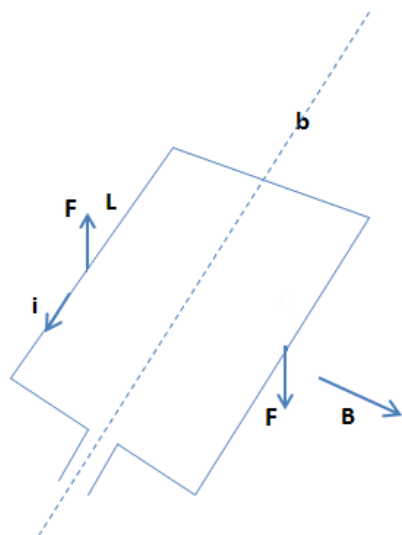
4.2 Elmotorer

För att kunna utföra sitt arbete behövde utrustningen innehålla någon typ av motor. Nedan följer en beskrivning av de alternativ som behandlades i projektet.

4.2.1 Likströmsmotorn

Grundprincipen för likströmsmotorn är att utnyttja magnetismens attraktiva kraft mot den motsatta polen genom att utnyttja en permanentmagnet tillsammans med en eller flera elektromagneter bestående av spolar.

För att få ut ett vridande moment läggs spolen vinkelrätt mot magnetfältet, om linningen i spolen är centrerad runt en axel och det går en elektrisk ström igenom den så bildas ett positivt och ett negativt magnetfält på var sin sida om axeln vilket ger upphov till en rotationskraft. Genom att endast se på en lindning går det att beskriva som i Figur 4.2.



Figur 4.2: Samband mellan ström, magnetfält och kraftens riktning i spolen på i en likströmsmotor.

Där b och L står för spolens bredd respektive längd, i står för ström, F för kraft och B för magnetfältets riktning. Som bilden visar så finns det en kraft F på vardera sida om axeln (den sträckande linjen) vilket bidrar till rotationen, dessa krafter uppkommer på grund av det elektriska polerna som bildas i spolens lindning. Lindningen strävar efter att lägga sig längst med magnetfältet och roterar för att uppnå detta tillstånd. Genom att byta håll på strömmen innan lindningen har nått sitt önskade läge roterar den ett lite till och genom att ständigt växla riktning på strömmen vid rätt tillfälle så roterar motorn.

Kraften F som utvinns beror på tre faktorer nämligen strömstyrkan " i " spolens längd " L " och magnetfältet " B " och momentet på kraften " F ", hävarmen " b " och " n " antal varv i spolen. Hävarmen blir " b " lång eftersom axeln går genom mitten på spolen och det blir en kraft uppåt

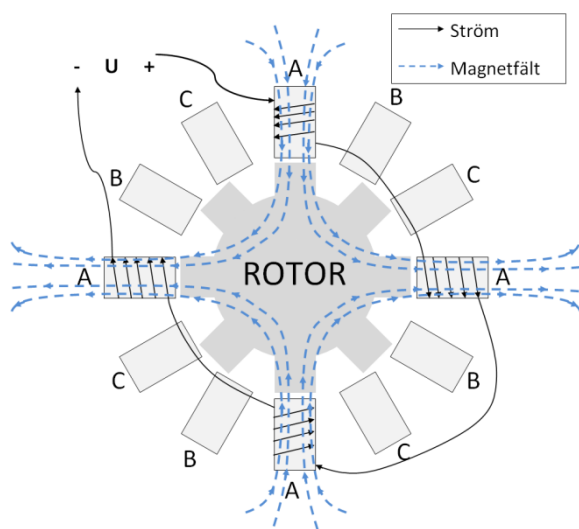
och en neråt enligt Figur 1.1. Antalet varv i spolen är direkt proportionellt momentet som utvinns från motorn så dubblas antalet lindningsvarv dubblas momentet. (Osbeck, 2011)

4.2.2 Stegmotorer

Stegmotorer kan nästan ses som digitala maskiner. Till skillnad från övriga elmotorer så arbetar de inte efter uppgiften ”rotera medsols”, utan istället ”rotera till position x”. Det största användningsområdet är därför positioneringssystem av olika slag.

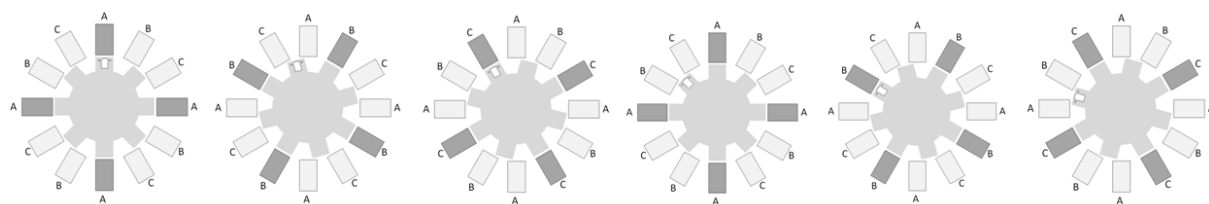
Generellt om stegmotorer

Statorn i en stegmotor kan se olika ut – i vissa fall används en permanentmagnetiserad stator och i andra fall inte – men funktionsprincipen är densamma. Motorns rotation drivs av ett slags magnetiskt kugghjul, där det yttre kugghjulets, det vill säga statorns, tänder består av elektromagneter som drar till sig de inre tänderna som sitter på rotorn. Nedan i Figur 4.3 illustreras en stegmotor utan permanentmagnet som med hjälp av omgivande elektromagneter dras till läge A. (Kuo, 2014)



Figur 4.3: Strömlindning och magnetfält för en stegmotor då lindning A är aktiv. Lindningarna för B och C visas inte här, men ser ut som för A.

I exempelmotorn ovan i Figur 4.3 visas bara lindningarna för elektromagneterna märkta A, men principen är densamma för B och C, vilket gör att man rotera motorn genom att koppla till och från strömmen till de olika lindningarna. Då strömmen slås på i lindning B så kommer alltså magneterna märkta med B dra till sig närmaste tänderna hos rotorn och så vidare. Detta förlopp illustreras nedan i Figur 4.4, där man genom att växla aktiv lindningskrets enligt mönstret A, B, C, A, B, C får rotorn att rotera motsols.



Figur 4.4: Stegningsförloppet hos en stegmotor. Aktiva lindningar visas som mörkgråa.

Varje stegmotor har en fast stegupplösning som beror på dess konstruktion. Den enkla exempelmotor som visas i Figur 4.3 och Figur 4.4 har en rotor med åtta tänder och tre faser att växla mellan, vilket gör att steglängden blir $\frac{360^\circ}{8 \cdot 3} = 15^\circ$. Vill man rotera motorn 75° , får man alltså stega motorn fem gånger i önskad riktning. Detta gör det möjligt att på ett väldigt enkelt sätt ta fram en *beräknad position*, vars pålitlighet är god under följande förhållanden:

- Motoraxelns startposition måste vara känd
- Yttre påverkan på axeln får inte vara för hög

Skulle motoraxeln påverkas av yttre vridmoment som är för stora så uppstår ett problem – de aktiva lindningarna drar till sig de närmaste tänderna hos rotorn, vilka tänder det är kan inte avgöras. En motor med åtta rotortänder har därför åtta möjliga positioner att hamna i då en viss lindning aktiveras, med mellanrum på 45° . Det mekaniska kugghjulet har alltså en risk att ”hoppa över kuggar”, lite som en cykelkedja, vilket kan leda till att en rotation på 75° i själva verket kan bli 30° – eller rent av 15° i fel riktning, om två kuggar skulle hoppas över. Positioneringssystem med höga precisionskrav och/eller stor yttre påverkan kan därför inte förlita sig på stegmotorns beräknade position.

För att öka stegupplösningen hos motorn, kan flera lindningar aktiveras samtidigt – om både lindning *A* och *B* är aktiva, så kommer rotorns tänder dras till ett läge mitt emellan *A* och *B*, en metod som kallas *halvsteg*. Detta går också vidareutveckla genom att köra de elektromagneterna på olika hög effekt, och på sådant sätt dra rotortänderna lite närmare *B* än *A*, och vice versa. Detta kallas för *mikrostegning* och är ett sätt att höja stegmotorns upplösning avsevärt. Det blir dock inte lika effektivt som att höja stegupplösningen på riktigt, eftersom fullstegen ger betydligt högre hållmoment än övriga positioner.

Trinamic PANdrive

I projektet användes ett stegmotorsystem ifrån tillverkaren Trinamic, som kallas PANdrive. Detta system består av en stegmotor med tillhörande styrenhet som innehåller följande:

- Ett flertal seriella kommunikationsgränssnitt (USB, RS-485, CAN)
- En lägesgivare som mäter motoraxelns rörelse och anger dess verkliga position, med en upplösning på 1024 positioner/varv.
- Ett system som mäter motorns belastning för att bland annat kunna detektera motorstopp, kallat Stallguard. Systemet beräknar belastningen genom att jämföra den elektriska energi som flödar in i motorn med den som flödar ut från den.

Flera motorstorlekar finns att välja på, men gemensamt är att de har ett stegavstånd på $1,8^\circ$ och en mikrostegupplösning på 256 mikrosteg per fullsteg.

5 UTVECKLING AV HÅRDVARA

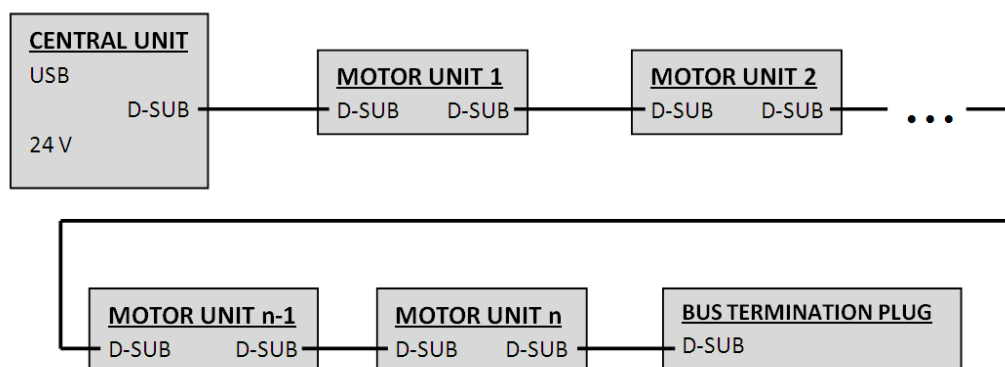
5.1 Val av motortyp

Det mest avgörande valet för hur systemet skulle byggas upp var vilken typ motor som skulle användas. Det gamla systemet byggde på en likströmsmotor – den var enkel att mäta belastning på, vilket gav en smidig och pålitlig kalibreringsprocedur, men också ganska benägen att slitas ut och sluta fungera. Om problemet satt i själva motorn eller i någon annan komponent var osäkert, det kan också ha varit växellådan eller potentiometern, som användes som lägesgivare, som begränsade livslängden.

Ett alternativ till detta skulle vara att bygga systemet runt en stegmotor. Detta kändes snabbt som ett lockande alternativ, då det skulle kunna leda till färre mekaniska komponenter som skulle kunna slitas ut. Det största problemet var att mäta belastningen, då stegmotorer inte har något direkt samband mellan driftström och vridmoment, så som likströmsmotorer. I en anslutande avdelning på SP arbetade Håkan Skoogh med ett antal stegmotorer av fabrikatet *Trinamic* i ett positioneringssystem. Han visade att dessa motorer hade ett inbyggt system för att mäta belastning, samt en beröringsfri lägesgivare. Efter en tids provkörning visade det sig att denna motor uppfyllde de krav som fanns på systemet, samtidigt som den gav en mycket enkel mekanisk konstruktion med minimalt slitage.

5.2 Uppbyggnad av systemet

Hårdvarusystemet består av en centralenhet, valfritt antal motorenheter och en slutkontakt, som kopplas ihop med 9-poliga D-sub-kontakter enligt Figur 5.1. Slutkontakten innehåller en resistans mellan de två kommunikationsledarna, vilket ska finnas i slutet av en RS-485 slinga. (RS-485 står för Recommended Standard 485 och är en gränssnittsstandard för seriell datakommunikation.) Vid sammankoppling av systemet ska alltså båda in- och utgångar vara inkopplade i samtliga motorenheter!



Figur 5.1: Huvudenheten och hårdvaruenheterna kopplas ihop i en RS-485-slinga med inbyggd strömmatning.

5.2.1 Centralenheten

Huvudlådan är en ganska liten låda som på ingångssidan har en ingång för 24 V ifrån en transformator och en USB ifrån datorn. Inne i lådan finns en gränssnittsomvandlare från USB till RS-485 och en D-sub-kontakt med 9 pinnar, dit både kommunikationen och 24 V

spänningen kopplas in. Eftersom RS-485 bara tar upp tre pinnar, så kopplas spänningen till samtliga sex kvarvarande pinnar för att ge lägre elektrisk resistans, och därmed minskat spänningsfall över kablaset.

5.2.2 Motorenheterna

Motorlådan är även den kompakt byggd, på sidan har den in- och utgångar i form av D-sub-kontakter, samt en lysdiod för indikering. Utgången används för att kunna koppla ihop en kedja av motorenheter och skickar bara vidare det som kommer till ingången. Till utgången på den sista motorn i kedjan ska slutkontakten kopplas in.

Inne i lådan sitter enbart stegmotorn med tillhörande styrkort, samt kablar till strömförsörjning, kommunikation och lysdioden. På framsidan sticker motoraxeln ut som kan kopplas till instrumentet via en Oldham-koppling.

5.2.3 Störningsmätning vid motorenheter

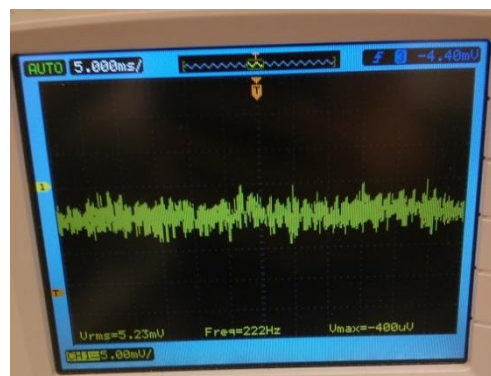
Utrustningen är tänkt att användas i mycket störningskänsliga miljöer och får därför inte skicka ut några nämnvärda störningar till omgivningen. Därför gjordes ett störningstest för att säkerställa detta.

Mätningar

Det första som letades efter var kapacitiva störningar. Mätningen gjordes i fyra steg genom att ha en referensmätning, se Figur 5.2, där utrustningen inte fanns i närheten, och sedan jämföra denna med ett antal mätningar precis vid en motorenhet.

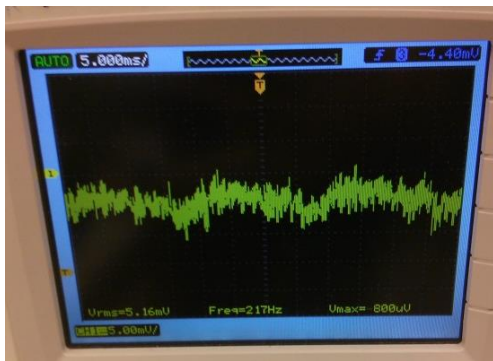


Figur 5.2: Resultatet ifrån referensmätningen.

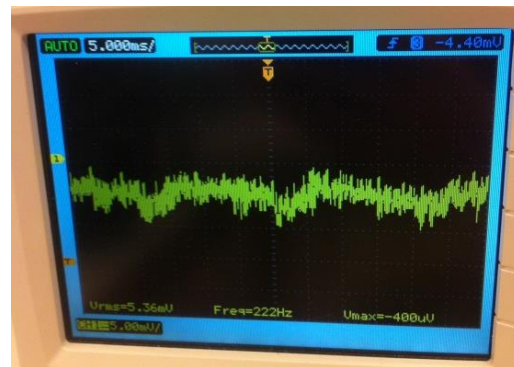


Figur 5.3: Kapacitiv störningsmätning vid en motorenhet utan strömmatning.

Referensmätningen gav en störning med en rsm-spänning 5,8 mV vid en frekvens på 1kHz enligt Figur 5.2. Denna störning kom alltså från andra saker som alltid finns i omgivningen, t.ex. taklampor och datorskärmar. En mätning vid utrustningen när den inte har någon strömmaning inkopplad ger en rms-signal på 5,49 mV vid en frekvens på 46,5 Hz enligt Figur 5.3. Anledningen till att denna signal faktiskt är lägre än referensmätningen är troligtvis att den är uppmätt precis bredvid motorenheten, som då skuggar en del strålning från omgivningen.



Figur 5.4: Resultaten då utrustningen är strömmatad.



Figur 5.5: Resultaten då enheten körs.

När utrustningen är strömmatad fås en rms-störning på runt 5,16 mV vid frekvensen 217 Hz (Figur 5.5), vilket innebär ytterligare minskning jämfört med referensen. En trolig förklaring till detta är att motorenheten nu har en förbindelse till jorden. Nästa mätning gjordes då motorn arbetade och gav då en störning med en rms-spänning på ca 5,36 mV vid 222 Hz (Figur 5.6). Nu ökar alltså störningssignalen en aning, men ligger fortfarande under referensmätningen. När systemet kördes utan någon koppling till jord blev det en klart märkbar störning, se Figur 5.6. Denna tydliga svängning får sin frekvens ifrån elnätet (50 Hz).



Figur 5.6: Resultatet med mätning utan jord till datorn.

Slutsats utifrån mätningar

En fråga som ställdes innan mätningarna genomfördes var hur stora störningar som skulle orsakas av att ha ständig spänningsmatning till motorenheterna – om dessa var för stora så skulle spänningen behöva slås av mellan körningarna. Mätningarna visade dock att spänningsmatningen inte gav några mätbara störningar, klart mindre än belysning och annat som fanns i omgivningen. Därför bedömdes det onödigt att bryta strömmen. Detta ger också högre säkerhet i systemet eftersom lägesgivarna i motorerna tappar sin information då strömmen bryts. Det är dock viktigt att värddatorn som styr systemet är kopplad till jord, för att strömmatningen ska bli störningsfri!

5.3 Monteringsrigg för Fluke 792A

Riggen konstruerades som en platta med en justerbar motorställning som man ställer/skjuter Fluke 792A på plats mot motorn. Det finns även fästen så det är enkelt att spänna fast Fluke 792A eller någon likande enhet och det är därför motorställningen är justerbar kunna vara kompatibel med liknade enheter. För att koppla samman omkopplarens axel med motorns används en *Oldham-koppling*. Detta tillåter momentöverföring även om axlarna ligger lite förskjutna i förhållande till varandra (Shin, 2014).

6 UTVECKLING AV PROGRAMVARAN

Trinamics PANdrive-system kan antingen styras med interna program i styrenhetens minne, eller från en extern dator via seriell kommunikation. Eftersom utrustningen skulle ta emot instruktioner ifrån ett befintligt LabVIEW-program som kontrollerade hela mätutrustningen, ansågs den bästa lösningen vara att även utveckla motorns styrsystem i LabVIEW. Detta gav både en smidig integrering i nuvarande system och gjorde det enklare för de som arbetar med utrustningen att göra eventuella ändringar i framtiden, då de redan hade erfarenhet av tekniken.

6.1 Kommunikation

PANdrive kan kommunicera med hjälp av ett antal seriella gränssnitt. Vid kommunikation via USB eller RS-485 skickas instruktioner till motorn i form av en sträng på 9 bytes enligt följande:

Tabell 6.1: Instruktionspaketen som skickas till PANdrive.

Byte nr	Betydelse	Kommentar
0	Modulens adress	Används vid kommunikation via RS-485 för att identifiera rätt mottagare.
1	Instruktionsnummer	Varje instruktion (ex. "kör åt höger") har ett unikt ID.
2	Typnummer	Är ofta 0, men behövs ibland för att precisera en instruktion. Ett exempel på detta är Instruktionen "hämta axelparameter", som måste följas av vilken parameter som ska hämtas, förslagsvis position eller mål.
3	Motornummer/ minnesbank	Motornumret är alltid 0 när man bara arbetar med enstaka självständiga motorer. Vid extern styrning används bara minnesbank 0, övriga banker kan användas för avbrottshantering och variabelagring. Denna byte är alltså i detta ändamål alltid 0.
4-7	Värde	Många instruktioner skall innehålla ett värde. "Gå till position" måste till exempel skickas tillsammans med den önskade positionen. Instruktionsvärden kan vara stora och använder därför 4 byte.
8	Kontrollsumma	Summerar värdena från alla tidigare 8 bytes. Används för att verifiera att överföringen gått felfritt.

Det första som gjordes var att ta fram ett VI i LabVIEW som översätter alla kommandon som behövs för styrningen till instruktionspaket enligt Tabell 6.1. Så fort instruktionen har tagits emot så skickas ett svar tillbaka från modulen, som en bekräftelse på att kommandot har tagits emot. I detta meddelande finns även utrymme för ett svarsvärde att skickas, ifall ett sådant skulle ha begärts. Svaren följer precis som instruktionerna alltid en förbestämd mall som redovisas i Tabell 6.2

Tabell 6.2: Svarspaket från PANdrive.

Byte nr	Betydelse	Kommentar
0	Svarsadress	Används vid kommunikation via RS-485 för att adressera värddatorn.
1	Modulens adress	Modulen som skickar svaret identifierar sig med denna (via RS-485).
2	Status	Ex. 100 betyder OK, 1 betyder fel kontrollsumma o.s.v.
3	Instruktionsnummer	Anger instruktionen som tagits emot av modulen.
4-7	Värde	Innehåller eventuella värden som efterfrågats i instruktionen.
8	Kontrollsumma	Som ovan i Tabell 6.1. Summerar tidigare 8 bytes.

Tolkningen av svarssträngen bakades in i samma VI som skickar instruktioner. Därmed sköts all kommunikation mellan värddator och motorenhet genom detta VI, som tar in önskad instruktion och sedan anger vad svaret blev och om några fel inträffade. Kommunikationsporten (COM-porten) bevakas för att läsa av svaret så fort det anlät, men vanligtvis tar det ca 2 ms.

Eftersom kommunikation över USB och RS-485 följer samma mall så ser styrprogrammet likadant ut i båda fallen. Detta är något som utnyttjades via utvecklingsprocessen – till en början användes USB som var mycket enkelt att koppla in och börja arbeta med, men under processens gång gjordes en övergång till RS-485. Detta gjordes för att systemet enklare skulle kunna kontrollera flera enheter samtidigt, då gränssnittet stödjer upp till 32 mottagare kopplade i ett nätverk (Buchanan, 2000). RS-485 identifierar enheterna med deras moduladresser – vilket gör det enkelt att ge en viss motor önskad adress och sedan ha full kontroll över vilken enhet som anropas. USB-gränssnittet använder sig inte av moduladresserna, vilket leder till att man där måste ge varje motor en egen COM-port. Det ger alltså värddatorn en buss för varje ansluten motor, istället för att all kommunikation skickas ut i samma kedja. Därför ansågs det vara värt att investera i ett RS-485-gränssnitt.

6.2 Kalibrering

För att motorenheterna ska kunna kopplas till ett flertal olika omkopplare och inte vara skräddarsydda för en specifik modell, måste de kunna kalibrera sig mot den utrustning som de kopplas till. Målet var att ta fram en kalibreringsprocess som var så snabb, automatisk och framförallt så pålitlig som möjligt, men för att säkerställa kompatibilitet vilken vridomkopplare som helst lades även ett manuellt läge till.

6.2.1 Vridmoment

Motorn måste givetvis vara tillräckligt stark för att vrida omkopplaren den är monterad mot, men får inte vara så stark att den riskerar att skada något. Eftersom målet var att framställa en utrustning som kan hantera allt ifrån väldigt lätta till mycket tunga omkopplare, behövde motorns vridmoment kunna ställas in från fall till fall.

Det maximala vridmomentet som en stegmotor kan framställa är proportionellt mot den ström som den matas med. Hur stark ström som körs genom motorn är en ställbar parameter i den

använda stegmotorns styrsystem, vilket gör det möjligt att kontrollera motorns vridmoment genom mjukvaran.

Den automatiska kalibreringen består av ett VI som till en början ställer in en mycket svag ström till motorn och sedan försöker rotera den. Lyckas inte det, så höjer den strömmen och försöker igen. Då användaren startar processen får den välja vilken typ av omkopplare som är aktuell – om det är ett vred som går att rotera hur många varv som helst, eller om det finns två ändpunkter. I det förstnämnda fallet så fortsätter programmet öka strömstyrkan tills motorn lyckas rotera ett helt varv i båda riktningarna utan att få stopp. I fallet med ändpunkter så ökas strömmen först tills motorn lyckas vrida ca 10°, och efter detta börjar den kontrollera hur långt den lyckas ta sig i båda riktningarna. Om det tar stopp på samma positioner efter en strömökning, så anses detta vara omkopplarens slutpositioner och kalibreringen avslutas, annars ökas strömmen ytterligen och ändpunkterna jämförs igen. Ett flödesschema för funktionen finns att hitta i appendix.

För att avgöra om motorn har lyckats vrida axeln används PANdrivens båda positionsparametrar. Den ena bygger på att den räknar hur långt motorn har instruerats att rotera, vilket då anger var den *borde* befinna sig, den andra är en lägesgivare som håller koll på hur axeln faktiskt rör sig och därmed anger en *korrekt* position. Genom att jämföra dessa två värden får man fram hur mycket motorn släpar efter, vilket kan användas för att detektera motorstopp. Detta används som en kontinuerlig säkerhetsfunktion i hela styrsystemet, men framförallt under kalibreringsprocessen. PANdrive har ett eget inbyggt system för motorstoppshantering, kallat StallGuard. Detta valdes bort på grund av att belastningen som systemet registrerar varierar kraftigt beroende på matningsströmmen, vilket gör att det måste finjusteras för varje belastningsscenario. Därför ansågs det säkrare att känna av motorstopp genom att mäta eftersläpning, då det gav en mer flexibel lösning som kan hantera att kopplas mot nya omkopplare med minimal ansträngning.

6.2.2 Stegkalibrering

Det automatiska systemet för att hitta omkopplarens olika steg bygger på att köra motorn från det ena ändläget till det andra och kontinuerligt mäta dess belastningsgrad. Om den aktuella omkopplaren är av typen som kan rotera obegränsat antal varv, så måste användaren ställa instrumentet till dess startläge och starta stegsökningen. Motorn kommer då rotera ett varv medurs och sedan tillbaka till startpositionen. Under tiden samlar programmet in information om i vilket läge motorn befinner sig och hur belastningen är i det läget. Om den aktuella omkopplaren däremot är av typen med ändlägen, så kommer systemet automatiskt leta fram dessa och sedan samla data i båda rotationsriktningarna.

Insamling av data

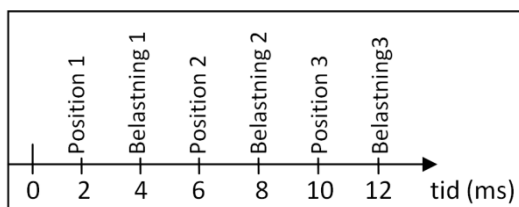
Datainsamlingen utförs genom att växelvis skicka förfrågningar om lägesgivarens position och motorns belastning till motorenheten. Hastigheten hos motorn valdes utifrån följande förutsättningar:

1. Den får inte vara för låg, för då fungerar inte belastningsmätningen.
2. Det förekommer en fördröjning på ca 2 ms vid varje förfrågan om ett värde från motorenheten (se avsnitt 6.1), vilket leder till att en hel samplingscykel tar ca 4 ms.
3. Eftersom lägesgivaren registrerar 1024 positioner/varv och det är önskvärt att ta tillvara på samtliga för att uppnå högsta möjliga precision, måste systemet hinna utföra 1024 samplingscykler/varv.

Resultatet blev att hastigheten enligt punkt 2 och 3 inte fick vara högre än ca 0,244 varv/s och enligt punkt 1 skulle vara så hög som möjligt. Ett snabbt test visade att 0,24 varv/s var tillräckligt hög hastighet för att belastningsmätningen skulle fungera, så det var hastigheten som valdes.

En annan effekt av fördröjningen i kommunikationen med motorenheten är att positionen som sägs höra ihop med en viss belastning är egentligen positionen strax *innan* den belastningen. En illustration av detta finns nedan i Figur 6.1. Eftersom motorns hastighet under dessa mätningar är ca 0,24 varv/sekund, uppstår ett positionsfel på omkring $0,17^\circ$.

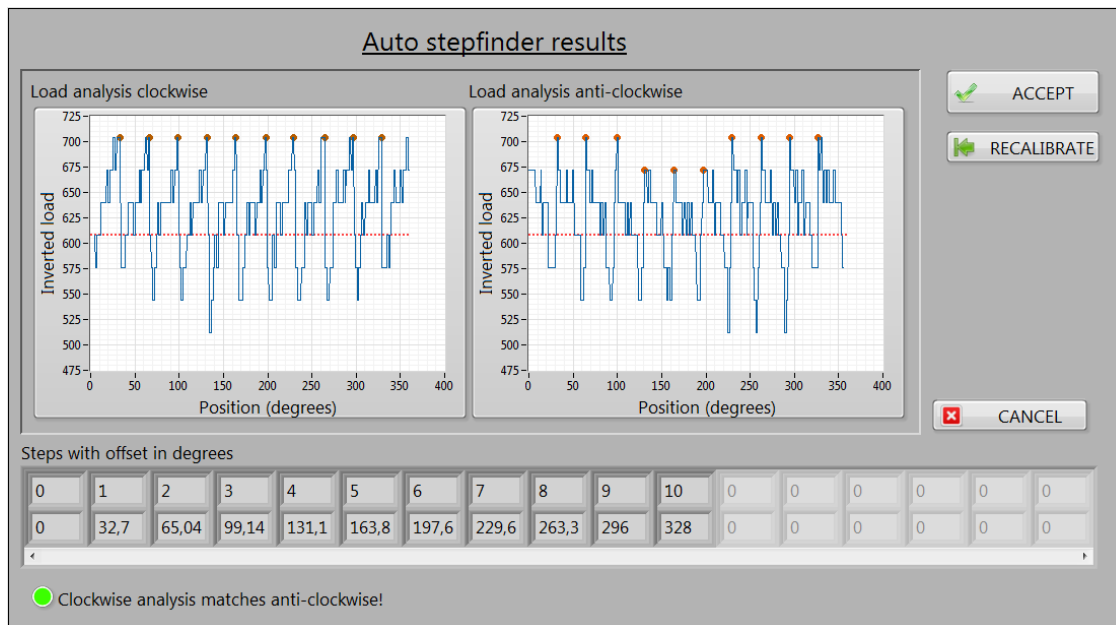
Detta fel elimineras dock av att mätningarna görs i båda rotationsriktningarna, som då ger lika stora fel åt var sitt håll.



Figur 6.1: Positions- och belastningssampling

Tolkning av data

Motorns hastighet är alltså vald för att ge strax över 1000 samplingar per varv, som nämndes i föregående avsnitt. Dessa värden ska sedan analyseras för att avgöra var omkopplarens steg befinner sig. Belastningsvärdet är inverterat, det vill säga ju högre värdet är desto mindre belastad är motorn. För att hitta stegen där omkopplaren ”klickar i” sina steg, måste man hitta de positioner där lasten är som minst – alltså eftersöks belastningsvärdenas maxpunkter.



Figur 6.2: Belastningsmätningar i båda riktningarna. Färdriktningen för motorn är åt höger på den vänstra bilden och åt vänster på den högra.

Ett antal försök gjordes för att ta fram den mest noggranna och pålitliga algoritmen för att hitta dessa maxpunkter. Eftersom mätningen är digital finns det en begränsad upplösning att ta hänsyn till, vilket gör att det sällan finns en entydig maxpunkt – värdet kan antingen flimra mellan två nivåer eller ligga konstant på max under en längre tid. Det gör att maxpunkten blir ett brett område och att man kan välja att antingen pricka ut områdets början, mittpunkt eller slut. Som synes ovan i Figur 6.2 så följs alltid toppvärdena av en distinkt nedförsbacke, en tydlig punkt när omkopplarens steg har nåtts och det börjar gå trögt igen. Att hitta den punkten, det vill säga maxområdets slutpunkt, visade sig ge mest pålitliga resultat.

För att hitta starten på den stora nedförsbacken går programmet igenom ett belastningsvärde i taget och letar efter en punkt som uppfyller följande krav:

1. Nästkommande värde är mindre
2. 10 samplingar bort har värdet sjunkit minst två steg (utesluter flimmar mellan två nivåer)
3. Under de kommande samplingsintervallen ökar aldrig värdet innan det nått medelbelastning (kontrollerar att backen är tillräckligt djup för att betraktas som ett steg)

Jämförelsen som nämns i punkt 2 innebär att omkopplaren inte får komma för nära nästa steg under de kommande 10 samplingsintervallen, vilket motsvarar en sträcka på ca 3,5°.

Avståndet mellan stegen måste alltså vara större än så för att systemet ska fungera och bör vara större än 6-7° för att alla steg ska hittas. Detta ansågs inte vara ett problem, då samtliga omkopplare som arbetades med under projektet har stegavstånd som är större än 25°.

Medelbelastningen som nämns ovan i punkt 3 sätts till belastningsvärdet mitt emellan max. last och min. last, och representeras i Figur 6.2. Då programmet har hittat ett max. belastningsvärde så väntar det tills belastningen har gått under och sedan tillbaka över detta värde innan det börjar leta efter ett nytt max. Detta för att undvika att samma steg hittas flera gånger.

Sammanslagning av de båda mätningarna

Som tidigare nämnt så samlas data in och tolkas i båda rotationsriktningarna. Efter det har programmet genererat två separata listor med hittade stegpositioner – en från medurs analys och en från moturs. Dessa två listor jämförs sedan i programmet – om de anses vara tillräckligt lika, så används medelvärden mellan de båda analyserna som stegpositioner. Om de inte anses vara tillräckligt lika så blir användaren varnad om detta och rekommenderas att starta en ny mätning.

Gränsen för att positionerna från de båda analyserna skulle anses likvärdiga var till en början satt till en maximal skillnad på 5°. Detta ledde till problem då tyngre omkopplare skulle styras – gränsen kunde då behöva höjas några grader. Så småningom gjordes dock systemet om så att det tog hänsyn till hur stora stegavstånd som mättes upp – den tillåtna skillnaden sattes då till ca 25 % av medelavståndet mellan stegen.

Testning av funktionen

Under utvecklingen utfördes kontroller för att se om de funna stegen stämde överens med verkligheten. Detta gjordes genom att för hand vrida omkopplaren till ett steg och känna efter att den låg på plats, för att sedan jämföra den positionen med positionen som den automatiska steghittaren angivit. Endast kalibreringar som mjukvaran godkänt, alltså där de två analyserna stämde överens, testades. Det som konstaterades var att den första prototypen lyckades hitta samtliga steg med ett genomsnittligt fel på 0,35° och ett maximalt fel på 0,7°, vilket ansågs vara fullt godkänt. Denna prototyp byggdes dock på en relativt liten och svag PANdrive-motor och kunde därför bara hantera lättvridna omkopplare.

Den andra prototypen byggdes på en större och starkare motor, eftersom det huvudsakliga målet med projektet var att automatisera ett mätinstrument (Fluke 792A) som hade en alldeles för trög omkopplare för den tidigare prototypen. Båda enheterna styrdes dock av samma mjukvara. Man lade märke till betydligt större skillnader mellan de två rotationsriktningarnas steganalyser och även att träffsäkerheten var något lägre. Efter ett antal justeringar blev felen dock aldrig större än ca 1,5°, och eftersom stegmotorn släpper axeln och låter den rotera fritt då den nått sin position så ”landade” omkopplaren i korrekt position under samtliga testkörningar. Samtliga justeringar som gjordes i programmet provkördes på båda prototyperna, kopplade till var sin omkopplare, för att åstadkomma ett styrsystem som var så allmänfungerande som möjligt.

6.3 Positioneringsprogram

Programmet som anropas för positionering av omkopplare tar in två instruktionsparametrar – omkopplarens namn och positionens namn. Det börjar med att söka igenom den information som sparats vid kalibreringen efter dessa två namn och skickar sedan en instruktion till motorn om att köra till tillhörande position. När motorn ger klarsignal, så dubbelkollar det att korrekt position har nåtts.

Positioneringsprogrammet innehåller också ett antal säkerhetsfunktioner. Den första går ut på att detektera strömbavbrott. Lägesgivaren som finns inbyggd i varje motor måste ha konstant strömförsörjning för att inte tappa sin position – skulle strömmen någon gång brytas så sätts aktuell position till 0. För att kunna känna av om detta har hänt arbetar programmet hela tiden med lite förskjutna positioner – det som i programmet benämns som positionen 0°, benämns av motorns lägesgivare som ca 360°. Detta betyder att om lägesgivaren någon gång har nollställts så ser programmet den aktuella positionen -360° och kan då larma för att positionen inte längre är korrekt.

Nästa säkerhetsfunktion går ut på att detektera motorstopp, hur detta går till rent tekniskt beskrivs i avsnitt 6.2.1. Utöver detta larmar också programmet då antingen något av namnet på omkopplare eller läge inte gick att hitta bland aktuella inställningar.

Skulle något av dessa larm utlösas, så pausas programmet och ett varningsfönster visas på skärmen som anger vad som gick fel, annars skickas en klarsignal ut som anger att önskad position har nåtts.

6.4 Engångsinitiering av PANdrive-enheter

Motorenheterna innehåller omodifierade PANdrive-motorer, som bara kompletterats med ett skyddshölje, en indikeringslampa och ett par D-sub-kontakter. För att programvaran ska kunna kommunicera med motorerna måste dessa vara inställda på rätt sätt, vilket löstes genom att ladda över ett initieringsprogram till PANdrivens interminne. Detta program körs igenom varje gång motorn kopplas till strömförsörjning och laddar då motorn med följande inställningar:

- Minimal strömstyrka då motorn är passiv, samt minimal väntetid innan motorn blir passiv då den nått sitt mål – detta för att låta omkopplaren ”falla ner” i sitt steg då motorn har jobbat färdigt.
- Hastighet och acceleration ställs in för att kalibreringen ska fungera optimalt (se ovan i 0).
- Stallguard ställs in för att ge snabb responstid på ändringar i belastning.
- Motorn tilldelas en moduladress som ställs in av användaren – varje inkopplad motorenhet måste ha en unik adress!
- Baud rate för RS-485 sätts till 115 200 bitar/sekund – nödvändigt för fungerande kommunikation mellan styrprogrammet och motorn! Hastigheten behöver också vara så här snabb för att den automatiska stegletaren ska fungera.

Ett VI togs fram för att ladda en PANdrive med detta program, för att på ett snabbt och enkelt sätt konfigurera nya motorenheter för kompatibilitet med systemet. Detta VI kan även användas då man vill ändra moduladressen till en enhet, till exempel om man skulle ha flera enheter med samma adress.

6.5 Anpassning efter styrande mjukvara

Då ett färdigt styrprogram hade tagits fram och skulle sammanfogas med det existerande LabVIEW-systemet på SP upptäcktes ett problem. LabVIEW har både begränsad fram- och bakåtkompatibilitet, vilket ledde till att de utvecklade VI inte gick att öppna i datorn som utför mätningarna. Detta konstaterades att det nog var ett vanligt scenario, då det är en stor operation för ett företag att uppdatera till nya mjukvara som inte är kompatibel med tidigare utvecklade program. Lösningen blev att kompilera en ".exe"-version av programmet, som kan köras på datorer som saknar kompatibelt LabVIEW-system, förutsatt att LabVIEW-run-time av tillräckligt sen version installeras.

7 ANVÄNDNING AV SYSTEMET

Nedan följer information och beskrivningar som kan vara till nytta vid användning av det system som tagits fram.

7.1 Bra att veta om systemet

För att motorenheter ska fungera i systemet behöver de någon gång ha laddats med ett initieringsprogram. Mer om detta finns att läsa nedan, i avsnitt 7.5. Nedan följer en snabb lista över annat som kan vara bra att känna till.

7.1.1 Ihopkoppling av hårdvara

- Motorenheterna kopplas i en kedja med centralenheten som startpunkt och en slutkontakt i slutet, som beskrivs i början av kapitel 0. Denna kedja bör ej innehålla mer än 32 motorenheter.
- Om standardkablar används, så fås ett spänningsfall på ca 0,1 Volt per meter kabel, vid höga effektkrav bör därför kabellängden inte överskrida 15 m.

7.1.2 Kalibrering efter omkopplare

- Då ett nytt instrument har kalibrerats så rekommenderas att en fil sparas med dess inställningar. Då behövs ingen ny kalibrering nästa gång instrumentet ska användas, utan det räcker att ladda in inställningsfilen!
- Systemet kan lära sig omkopplarens steg på två sätt:
 - Automatisk stegkalibrering fungerar som bäst då stegavstånden hos omkopplaren är 10° eller mer och alla steg är ungefär lika tröga.
 - Manuell stegkalibrering låter användaren vrida omkopplaren, antingen för hand eller med motorn, och spara önskade positioner manuellt. Detta möjliggör kalibrering av i princip vilka vridreglage som helst!
 - Oavsett vilken metod som används så kan stegpositionerna efterjusteras. Mer om detta följer i avsnitt 0.
- Även vridmomentet kan ställas in automatiskt eller manuellt:
 - Det automatiska systemet testas sig fram för att hitta den lägsta vridkraft som behövs, och lägger sedan på en liten marginal. Under denna kalibrering förekommer en del ryck och vibrationer i axeln, därför bör den ej användas mot extremt vibrationskänsliga instrument!
 - Manuell inställning görs genom att dra på ett reglage på skärmen. En knapp kan tryckas in för att prova inställningen genom att köra motorn.
 - Precis som stegpositioner kan även vridmomentet justeras i efterhand.
 - Vid användning mot omkopplare med oregelbunden tröghet rekommenderas manuell justering av vridmomentet – antingen genom en helt manuell vridmomentsinställning eller genom en efterjustering. Detta för att minska risken för motorstopp under drift.

7.1.3 Styrning av omkopplare

- Efter att en enhet har kalibrerats är det viktigt att dess strömmatning aldrig bryts! Händer detta så tappas enheten sin position och ger ett felmeddelande då den beordras att flytta till en position. Då krävs att enheten kalibreras på nytt för att den ska fungera!
- Då omkopplaren har ställts till önskad position, släpper motorn nästan hela sitt hållmoment. Genom yttre påverkan kan alltså dess position ändras, utan systemets vetskap!

7.2 Mjukvarans komponenter

Den utvecklade mjukvaran består av tre program:

1. **Rotary_switch_settings.exe** – Används för att kalibrera mot omkopplare.
2. **Rotary_switch_move.exe** – Används för att köra en omkopplare till önskat steg.
3. **Rotary_switch_install_motor_unit.exe** – Används för att initiera en PANdrive-motor för användning i systemet, samt för att ställa in adresser till motorenheter (se 6.3).

De två första programmen finns både som fristående program och som VI:s. Om de fristående versionerna används så krävs det att de båda programmen är placerade i samma mapp! Detta på grund av att alla inställningar som görs sparas i en fil i programmappen, som positioneringsprogrammet sedan måste hitta. VI-versionerna har inte detta krav, här fås istället inställningarna som utdata ifrån inställningsVI:et, som sedan får sparas som en variabel som matas in som indata i positioneringsVI:et.

7.3 Kalibrering av moduler

Då *Rotary_switch_settings.exe* körs för att kalibrera inkopplade moduler visas ett fönster enligt Figur 7.1 där den aktuella informationen visas. Vid start av programmet skall det kontrolleras att rätt COM-port är inställd och efter det görs en sökning efter moduler genom att trycka på knappen ”Scan modules” längst upp till vänster bredvid COM-portsinställningen.

The screenshot shows the 'Rotary Switch Settings' window. At the top left, there is a 'VISA resource name' dropdown menu set to 'COM9'. To its right is a 'Scan modules' button. Below this is a table with three columns: 'Position nr:', 'Position: (in degrees)', and 'Name:'. The table contains 15 rows, all with '0' in the first two columns and empty in the third. To the right of the table is a configuration section with fields for 'Address: 0', 'Type: Unknown', 'Torque: 0', and 'Number of steps: 0'. Below these fields are buttons for 'Load module config', 'Configure new instrument', and 'Save current module config'. Further down is an 'EDIT CONFIGURATION' section with 'Edit torque' and 'Edit steps' buttons. At the bottom right are 'UNDO CHANGES' (with a red X icon) and 'ACCEPT' (with a green checkmark icon) buttons. The 'NATIONAL' logo is visible at the bottom center.

Figur 7.1: Konfigurationspanel.

När scanningen är slutförd kommer ett fönster att visas med hittade moduler där de skall namnges för att identifiera en modul finns en identifiera knapp ”Identify” för varje modul som får en lampa att blinka på den aktuella enheten se Figur 7.2.

The screenshot shows the 'Set Module names' window. It has a table with two columns: 'Address:' and 'Name:'. The first row has '1' in the address column and 'Module_1' in the name column. The second row has '12' in the address column and 'Fluke 794A' in the name column. There are five more rows with '0' in the address column and empty in the name column. Each row has an 'Identify' button to its left. At the bottom right is an 'ACCEPT' button with a green checkmark icon.

Figur 7.2: Panel för namngivning av moduler.

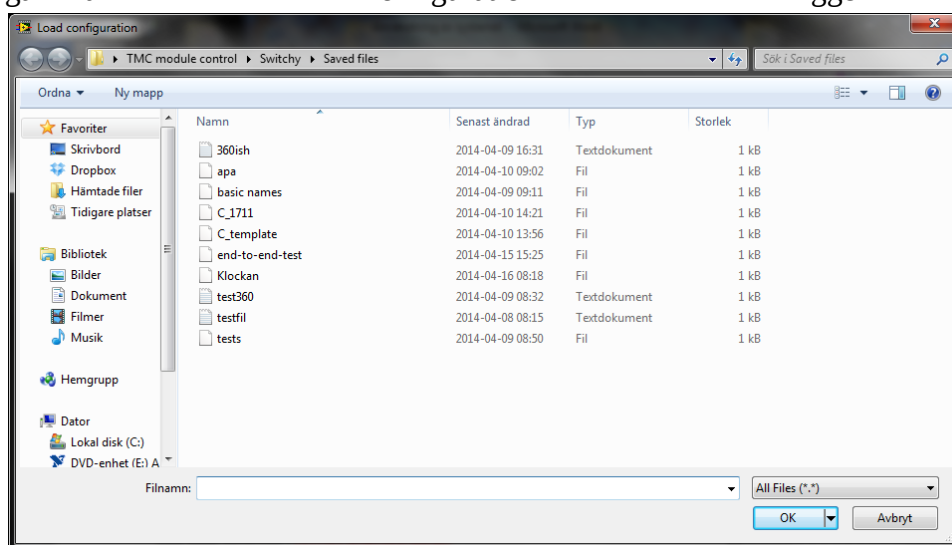
När scanningen är slutförd så kommer den första modulen att visas längst upp till höger i konfigureringsfönstret och dess inställningar visas se Figur 7.1. Genom att trycka på den lilla

pilen i den rutan kommer det upp en lista med de hittade enheterna så det enkelt går att växla mellan dem för att se deras inställningar.

När modulerna har identifierats och namngivits skall det ställas in efter önskat instrument och detta kan göras på två olika sätt genom att trycka på någon av följande knappar. ”Load module config” för att ladda in en gammal konfiguration eller ”Configure new instrument” för att göra ny Konfigurering.

7.3.1 Ladda konfiguration från fil

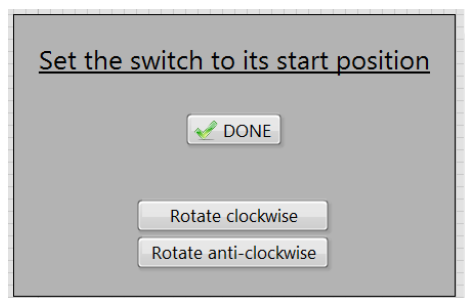
Om ” Load module config” väljs så kommer det upp ett fönster se Figur 7.3 där önskad gammal konfiguration ligger sparad.



Figur 7.3: Exempel på hur det kan se ut när en fil laddas.

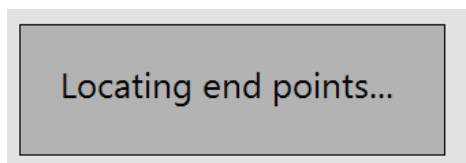
När filen är vald så kontrolleras det att den är i rätt format och säger till om något är fel och om filen är godkänd så går programmet vidare till att hitta startpunkten. Det görs genom att öppna ett nytt fönster som är beroende på om det är en ”End-to-end with steps” eller ”360 degree with steps” omkopplare.

Är det ett ”360 degree with step” instrument som har valts kommer det upp ett fönster se Figur 7.4. Där du beds att sätta startpositionen manuellt och detta kan antingen göras för hand eller genom att köra motorn med knapparna i rutan ”Rotate clockwise” för medurs och ”Rotate anti-clockwise” för moturs. När motorn står i startposition så trycker man på ”DONE” och då sparas startpunkten.



Figur 7.4: Panel för startposition för ett ”360 degree with steps” omkopplare.

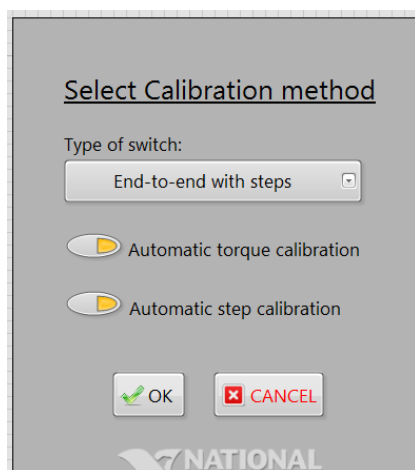
Om det istället är en "End-to-end with steps" omkopplare så letar den automatiskt ändlägena och undertiden visas ett fönster, se Figur 7.5.



Figur 7.5: Bild som visas när ändlägena hittas för "End-to-end with steps" omkopplare.

7.3.2 Kalibrera nytt instrument

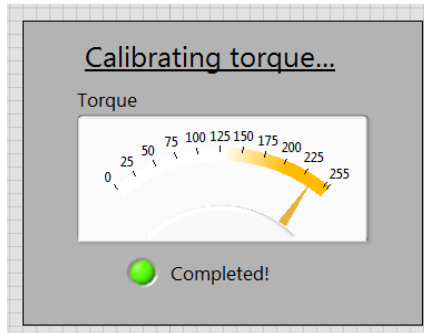
Om "Configure new instrument" väljs öppnas ett nytt fönster där några val skall göras enligt Figur 7.6. Det första är vilken typ av omkopplare det är och det finns följande alternativ: "End-to-end with steps" och "360 degree with steps". Val två och tre ställer in om det skall göras en automatisk eller manuell kalibrering av vridmoment och stegen. Eftersom vridmomentkalibreringen och stegkalibreringen är skilda ifrån varandra så går det köra den ena automatisk och den andra manuellt och vice versa.



Figur 7.6: Panel där inställningar görs för en nykalibrering.

Automatisk vridmomentskalibrering

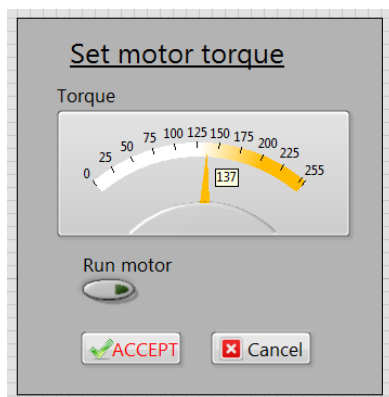
När automatisk vridmomentskalibrering väljs börjar motorn att köra med lågt vridmoment och ökar sakta tills den anser att det är tillräckligt för instrumentet och slutligen lägger på en liten säkerhetsmarginal. Förloppen studeras live enligt Figur 7.7.



Figur 7.7: Visar vridmomentsökningen vid automatisk vridmomentskalibrering.

Manuell vridmomentskalibrering

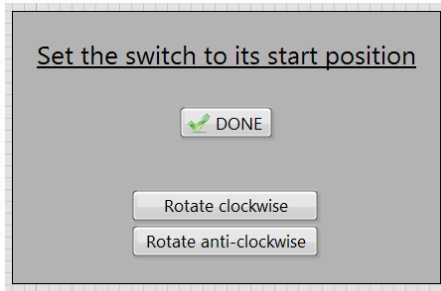
Om manuellt läge har valts för vridmomentskalibrering kommer ett nytt fönster upp enligt Figur 7.8 där justeringen av vridmoment sker genom att dra i visaren till önskad nivå. Det går även att testköra motorn genom att trycka på knappen "Run motor". När kalibreringen är färdig sparas värdet genom att trycka på accept och väljs avbryt behålls det gamla värdet.



Figur 7.8: Pilen dras till önskat vridmoment vid manuellt vridmomentskalibrering.

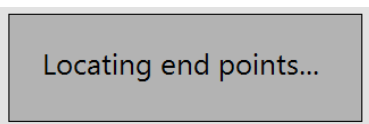
Automatisk stegkalibrering

När vridmomentkalibreringen är gjord är det första steget i automatisk stegkalibrering att ta fram startpunkten. Är det ett "360 degree with step" instrument som har valts kommer det upp en ruta se Figur 7.9. Där du beds att sätta startpositionen manuellt och detta kan antingen göras för hand eller genom att köra motorn med knapparna i rutan "Rotate clockwise" för medurs och "Rotate anti-clockwise" för moturs. När motorn står i startposition så trycker man på "DONE" och då sparas startpunkten.



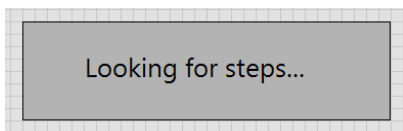
Figur 7.9: Panel för att sätta startposition för ett "360 degree with steps" instrument.

Är det istället ett "End-to-end with steps" instrument så letar den automatiskt ändlägena och undertiden visas ett fönster, se Figur 7.10.



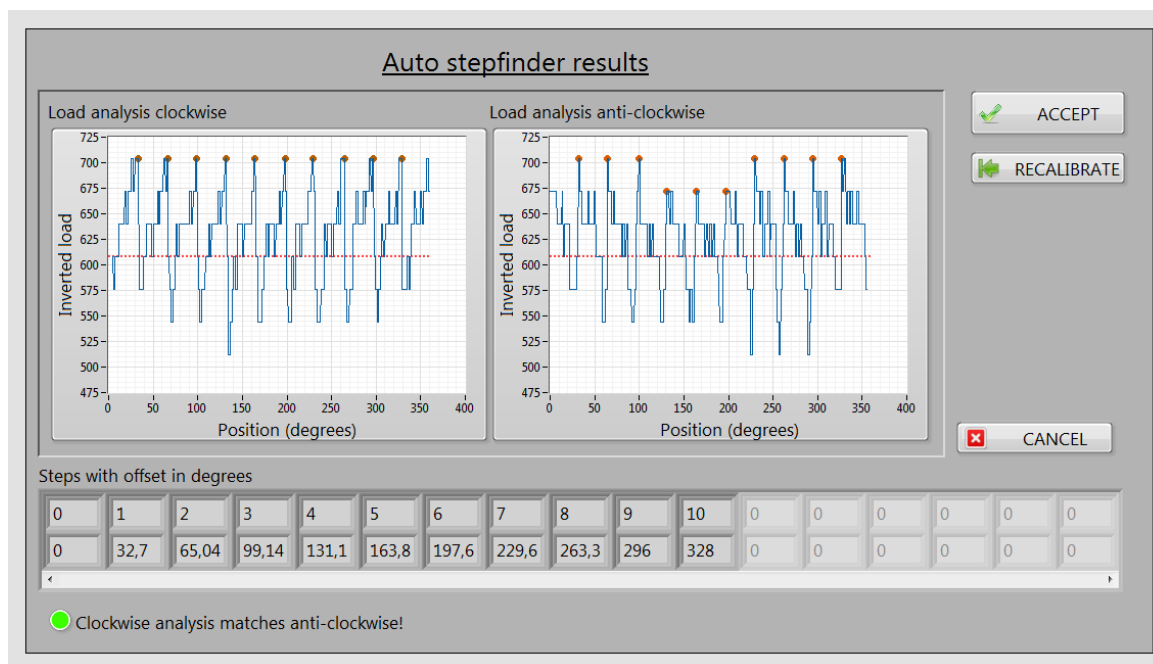
Figur 7.10: Bild som visas när ändlägena hittas för "End-to-end with steps" instrument.

Efter att startpunkten eller ändlägena är satta påbörjas stegkalibreringen samtidigt som ett fönster öppnas för att visa att den arbetar se Figur 7.11.



Figur 7.11: Bild som visas när den letar steg.

När stegkalibreringen är färdig kommer det upp ett nytt fönster se Figur 7.12 med två grafer som visar var sin momentkurva med stegen utpunktade, en för medurs och en för moturs. Det finns även en lampa med tillhörande text som anger om de båda mätningarna gav matchande resultat. Om det indikeras att mätningarna ej stämde överens, bör en omkalibrering utföras då någon av positionerna kan vara felaktig! Dessutom visas knappar för att godkänna (ACCEPT), kalibrera om (RECALIBRATE) och avbryta (CANCEL) för att behålla den ursprungliga stegkalibreringen.



Figur 7.12: Presenterar resultatet ifrån den automatiska stegkalibreringen.

När man godkänt så kommer man till ett namngivningsfönster se Figur 7.13 där namngivning av stegen sker, det går även att ladda och spara namnen med hjälp av "SAVE" och "LOAD" knapparna. ". Filen som laddas in kan antingen vara en ren namn-fil som skapas genom att trycka på "SAVE" knappen eller en gammal kalibreringsfil med samma antal steg.

Name Editor

Steps: Names:

0	Step_0
1	Step_1
2	Step_2
3	Step_3
4	Step_4
5	Step_5
6	Step_6
7	Step_7
0	
0	
0	
0	
0	
0	

SAVE/LOAD NAME TEMPLATE

SAVE LOAD

ACCEPT

Figur 7.13: Fönstret där namngivning av steg sker.

Manuell stegkalibrering

Om manuell stegkalibrering har valts öppnas ett fönster se Figur 7.14 där punkter läggs till och tas bort i turordning genom att trycka på knapparna "Add" eller "Remove". Motorn kan antingen vridas för hand eller köras med knapparna "Rotate clockwise" eller "Rotate anti-clockwise" för att komma till önskad position. Observera att steg 0 är startpunkt. När alla steg är satta ska de namnges och det går att göra antingen genom att skriva in dem eller genom att ladda in en fil med namn genom att trycka på knappen "LOAD". Filen som laddas in kan antingen vara en ren namn-fil som skapas genom att trycka på "SAVE" knappen eller en gammal kalibreringsfil med samma antal steg. Genom att trycka på "ACCEPT" knappen godkänns stegen och namnen och kalibreringen är färdig.

Manual Step Adder

Step: Location: Name:

0	0	22 mV
1	34,8	Step_1
2	76,29	Step_2
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	

ADD STEP POSITIONS

Current position: 76,29 Add Remove

Rotate clockwise

Rotate anti-clockwise

SAVE/LOAD NAME TEMPLATE

SAVE LOAD

ACCEPT

Figur 7.14: Fönstret där man manuellt lägger till och ändrar steg.

Ändra vridmoment

Om "Edit torque" har valts öppnas fönstret för manuell vridmomentskalibrering (Figur 7.8 på sida 25). Se avsnitt 0 för att ändra vridmomentet.

Ändra steg

Om "Edit steps" väljs öppnas ett fönster (se Figur 7.15) där man kan ändra namn, manuellt uppdatera, lägga till och ta bort önskad position genom reglaget "Step" och knapparna "Add step", "Remove step" och "Overwrite with current position". Motorn kan flyttas för hand till önskad position, köra den dit med hjälp av knapparna för att köra medurs eller moturs ("Rotate clockwise" eller "Rotate anti-clockwise") eller genom knappen "Move to step" för att köra till valt steg. Det går även att spara och ladda namn med knapparna "SAVE" och "LOAD". Filen som laddas in kan antingen vara en ren namn-fil som skapas genom att trycka på "SAVE" knappen eller en gammal kalibreringsfil med samma antal steg. De två sista knapparna är "ACCEPT" och "CANCEL" som är till för att spara respektive inte spara ändringarna.

Step	Location	Name
0	0,0	22 mV
1	28,1	220 mV
2	58,7	700 mV
3	88,6	2,2 V
4	118,5	7 V
5	148,0	22 V
6	178,2	70 V
7	207,1	220 V
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	

Figur 7.15: Fönstret där man efterredigerar steg och namn sker.

7.4 Positionsanrop

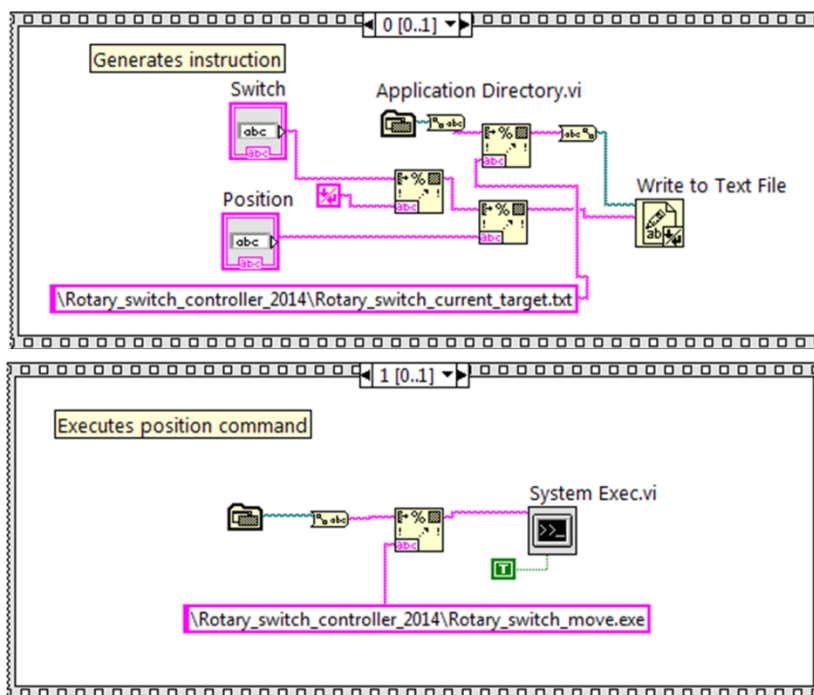
För att beordra omkopplaren att vridas till önskat steg anropas *Rotary_switch_move*. Detta program finns både som fristående program och som VI.

7.4.1 Fristående version

Det fristående programmet är beroende av att två filer finns i dess mapp:

1. Kalibreringsfilen som genererats av *Rotary_switch_settings.exe*.
2. En textfil med önskad position vid namn "*Rotary_switch_current_target.txt*". Textfilen ska bestå av två rader, där den första raden anger omkopplarens namn och den andra anger positionens namn.

Förutsatt att LabVIEW-run-time-engine finns installerat på datorn, kan systemet alltså styras utan själva LabVIEW-programvaran. Allt som behövs är att kalibreringsprogrammet körs och att den ovannämnda textfilen genereras innan positioneringsprogrammet anropas. Nedan i Figur 7.16 demonstreras hur funktionen kan anropas ifrån ett LabVIEW-program, vilket som bekant är det som användes i projektet. I exemplet har Rotary_switch-programmet placerats i en undermapp till det anropande programmet, vid namn "*Rotary_switch_controller_2014*". I första steget så skapas textfilen där variabeln "Switch" läggs på första raden och "Position" på raden under, sedan anropas den körbara filen som ställer omkopplaren i den beställda positionen.

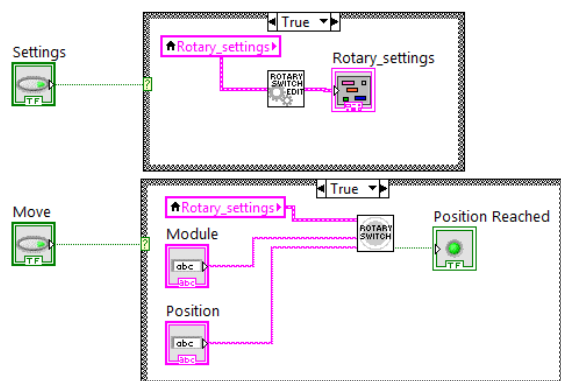


Figur 7.16: Programexempel som kör en omkopplare till önskad position med hjälp av den fristående versionen av positioneringsprogrammet.

Eftersom programmet är fristående så behöver alltså instruktionerna inte komma ifrån LabVIEW, det enda viktiga är att textfilen innehåller en korrekt instruktion.

7.4.2 VI-version

VI-versionerna av programmen sparar ingenting i filer, utan använder sig istället av in- och ut-data internt i LabVIEW. Nedan i Figur 7.17 visas ett enkelt exempel på ett program där inställnings- och positioneringsprogrammet kan anropas med respektive knapp. Önskad position anges, precis som i Figur 7.16, i två textsträngar.



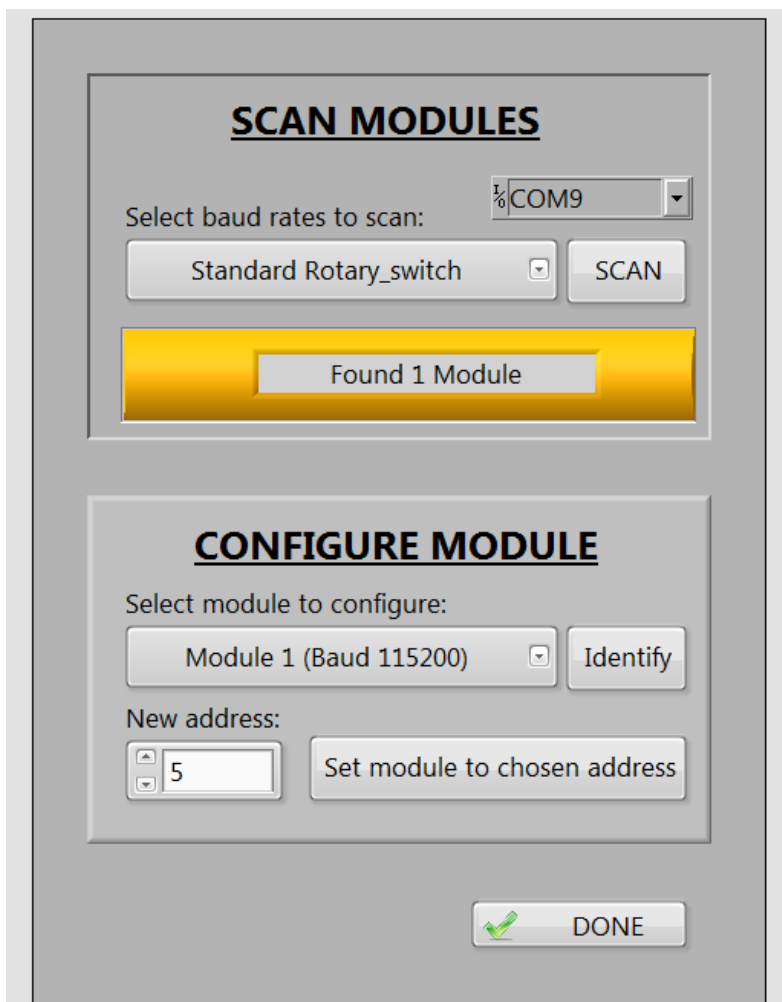
Figur 7.17: Programexempel på hur VI-versionen av positioneringssystemet kan anropas.

7.5 Installation av motorenheter

Programmet *Rotary_switch_install_motor_unit.exe* används för att anpassa PANdrive-motorer för användning i systemet, samt för att ge dem en önskad adress. Detta behöver egentligen bara utföras en gång för och samma motor, men moduladressen är praktiskt att kunna ändra vid ett senare tillfälle, då det inte får förekomma flera enheter med samma adress i ett system. Nedan, i Figur 7.18, visas en bild på hur detta program ser ut.

Under "Scan modules" kan användaren välja vilka baud rates som skall sökas igenom. Alternativen är PANdrives standard inställning (9 600 bitar per sekund), standardinställningen för *Rotary_switch*-enheter (115 200 bitar per sekund), eller samtliga hastigheter som en PANdrive kan ställas in på.

Då en genomsökning av enheter har gjorts, hamnar alla motorer som hittats under "Configure module". Här finns en knapp för att beordra den valda motorn att identifiera sig med sin lysdiod, samt en knapp som konfigurerar motorn för användning i *Rotary_switch*-systemet och tilldelar den adressen som användaren har angivit under "New address".



Figur 7.18: Användargränssnittet för att installera en PANdrive-motor eller ändra dess adress.

8 RESULTAT OCH DISKUSSION

I det stora hela ansågs resultatet av projektet vara lyckad, då den framtagna produkten fungerade väl och uppfyllde de krav som ställts av SP.

8.1 Uppföljning av målsättning

Fyra enheter har testats i drift parallellt utan problem. Mer än så har inte funnits möjlighet att provköra och inte heller funnits något krav på, men troligtvis fungerar betydligt fler. Rent teoretiskt så klarar den seriella bussen av 32 enheter innan någon form av repeater krävs. Den största begränsningen ligger snarare i att spänningsmatningen går igenom samma kabel, som är relativt tunn och ger ett spänningsfall på ca 0,1 Volt per meter. Skulle kabellängden överskrida 15 m så kan en tjockare kabel behövas beroende av hur tungt motorerna ska belastas.

Självkalibreringen fungerar mestadels som önskat, men är inte riktigt så allmänfungerande som önskat. Den automatiska stegkalibreringen fungerar näst intill felfritt för den primära utrustningen, det vill säga *Fluke 792A*, där samtliga steg hittas med hög snabbhet och precision. Vridmomentskalibreringen har inte visat sig riktigt lika pålitlig mot den utrustningen. Vid vissa tillfällen har ett steg misstolkats som en ändpunkt och ibland har ett vridmoment tagits fram som fungerat utmärkt under en lång tid innan ett plötsligt motorstopp inträffat. Mot en annan omkopplare som arbetades med under projektet (tillhörande en dekadresistor) hittades dock pålitliga vridmoment så gott som varje försök, så det skiljer från utrustning till utrustning. Rekommendationen är att finjustera vridmomentet manuellt för att säkerställa hög pålitlighet! Även stegkalibreringen har ett manuellt läge att använda om det automatiska stöter på problem, vilket gör att det alltid går att få en kalibrering med hög precision.

Livslängden är svår att utvärdera då produkten är nytillverkad, men förutsättningarna anses vara goda. Den föregående utrustningen innehöll släpande mekaniska komponenter som slets ner med tiden – en trolig orsak till dess begränsade livslängd var potentiometern som angav position, en annan kan ha varit själva motorn. Genom att använda en kontaktlös lägesgivare och en stegmotor utan växellåda har så mycket mekanik som möjligt eliminerats, vilket borde leda till kraftigt begränsat slitage. Det som kan tänkas ge upphov till problem är de vibrationer som uppstår vid motorstopp, då stegmotorn hoppar över steg. Detta ska dock bara uppstå under kalibreringen, så vid styrning av vibrationskänslig utrustning kan problem undvikas genom att utföra en manuell kalibrering för att undvika motorstoppen.

I övrigt så är utrustningen enkel att montera, förutsatt att en Oldham-koppling kan fästas på omkopplarens axel, och den ger inte ifrån sig några direkt mätbara elektriska störningar.

8.2 Arbetsgång

Det första som gjordes var att med hjälp av de önskemål och krav som SP tillhandahållit ta fram en kravspecifikation för att utifrån den ta fram några lösningsförslag. Kraven handlade mycket om vikten av störningsskydd, CE-märkning och att programmet skulle vara gjort i LabVIEW.

Val av motortyp var den fråga som spelade störst roll inför den fortsatta utvecklingen produkten eftersom den bestämde hur moment och läge skulle beräknas. Genom att jämföra för- och nackdelar mellan ett antal alternativ valdes en stegmotor med tillhörande kringutrustning ut för att användas till produkten.

Det fanns redan en stegmotor av den valda typen på SP, dock en variant med för lågt vridmoment för projektets huvudmål. Alla motorer av denna typ går dock styra med samma mjukvara, så den svaga motorn användes för att fastställa att det gick att få ut de data som var nödvändiga för att mäta moment och position och på så sätt kunna ta fram ett kalibreringsprogram. Det gjordes med hjälp av LabVIEW och en testtrigg. Samtidigt gjordes även störningsmätningar för att avgöra om motorn även uppfyllde detta kravet eller om det skulle behövas vidta åtgärder för att dämpa störningarna vilket den gjorde.

Efter störningsmätningarna mättes momenten ut för på de aktuella instrumenten för att kunna beställa en motor med rätt vridmoment, därefter vidareutvecklades LabVIEW- programmet som skulle kalibrera och styra motorn automatiskt för önskat instrument.

När motorenheten äntligen kom var LabVIEW-programmet i stort sett färdigt funktionsmässigt och en testtrigg byggdes för att kunna testa och se att motorn även fungerade i praktiken innan en prototypplåda togs fram. När två enheter eller fler skulle köras parallellt fungerade inte kommunikationen på ett önskvärt sätt så alternativa gränssnitt granskades och valet föll på RS-485 eftersom det var snabbast och ansågs mest lättanvänt.

När prototypplådan var färdig gjordes ett sista störningstest för att säkerställa att allt stod rätt till på den fronten innan den skickas iväg för CE märkning och efter det så fanns det lite tid över för att arbeta med sekundäruppgiften och ta fram ett LabVIEW-program för styrning av batterienheter för *Fluke 792A*

8.3 Återkoppling till tidsplanen

Tidsplanen som lades upp i början av projektet visade sig bli ganska bakvänd. Den mekaniska konstruktionen hade antagits ta en stor del av tiden, likaså den styrande elektroniken och strömförsörjningen. Sist av allt skulle ett styrprogram tas fram.

Detta ändrades dock efter en kort provkörning av PANdrive-systemet, som visade sig innehålla så gott som all hårdvara som behövdes för utrustningen – en tillräckligt stark motor, en lägesgivare och ett sätt att mäta motorbelastning. Efter denna upptäckt hade projektet i princip förvandlats till ett rent styrprojekt, där den enda hårdvaran som behövde arbetas med var skyddshöljen och elektriska gränssnitt.

Detta gjorde att mer tid kunde läggas på att ta fram en genomarbetad mjukvara och därmed ge ett säkert och stabilt system.

8.4 Allmän diskussion

Projektet har varit både intressant och lärorikt och under hela arbetets gång har vi haft bra hjälp att tillgå från personalen på SP då det har behövts. Vi använde LabVIEW för första gången då vi inledde arbetet och har sedan lyckats få god förståelse för systemet och utvecklat en ganska komplex mjukvara. Som vanligt då det arbetas med digitalteknik och datakommunikation så har det uppstått en del svårförklarade problem och fenomen, men i och med att vi hade så pass mycket tid att lägga på mjukvaruutvecklingen så har vi hunnit hantera alla dessa. Vi känner oss nöjda med det färdiga systemets funktion, flexibilitet och stabilitet!

Huvudmålet att styra roterande omkopplare med distinkta steg är uppfyllt, men systemets uppbyggnad tillåter fler användningsområden – med hjälp av den manuella stegkalibreringen så kan det hanteras i princip vilket positioneringssystem som helst, förutsatt att någon motor i PANdrive-serien kan driva det. Man skulle till exempel kunna styra sin radio med utrustningen genom att koppla frekvensvredet till en motorenhet och manuellt ställa in positionerna för ett antal kanaler. Telefoner med fingerskivor är en annan möjlighet. Dessa kanske inte är så troliga, eller ens praktiska alternativ, men de visar ändå på utrustningens mångsidighet. Att just PANdrive-motorer måste användas är såklart en begränsning, men denna serie innehåller motorer av många olika storlekar och vridstyrkor.

En möjlig vidareutveckling finns i fästnanordningen. Målet att enkelt kunna koppla motorenheterna till diverse mätutrustningar på SP uppfylldes, men det krävdes att dessa utrustades med Oldham-kopplingar. För att göra utrustningen mer mångsidig skulle en ny fästnanordning tas fram som har möjligheten att ”greppa” olika vred utan att kräva modifikation hos det den ska kopplas till.

9 SLUTSATS

Sammanfattningsvis var projektet lyckat. Systemet som utvecklades uppfyllde samtliga krav som ställdes:

- Utrustningen är enkel att montera mot omkopplare
- Flera enheter kan användas samtidigt
- Utrustningen ger inte ifrån sig störningar som kan påverka närliggande instrument
- Utrustningen kan självkalibrera sig mot inkopplat instrument

Livslängden har inte kunnat testas under den relativt korta projekttiden, men har alla förutsättningar att bli god, då antalet mekaniska kontaktytor har minimerats.

SP är mycket nöjda med den utvecklade utrustningen och planerar att sätta den i drift snarast! Eventuellt kommer de även sälja vidare till liknande organisationer i Europa.

REFERENSLISTA

- Buchanan, W. (2000). RS-422. RS-423 and RS-485. In *Computer Busses* (p. 223). Edinburgh: Napier University.
- Kuo, D. B. (2014, April 29). Stepper Motor. *AccessScience*.
- Morgan, D. (1994). *A handbook for EMC testing and measurement*. London: Peter Peregrinus Ltd.
- National Instruments. (n.d.). *www.ni.com*. Retrieved maj 20, 2014, from NI LabVIEW for Higher Education: www.ni.com/labview/applications/academic/
- Osbeck, M. (2011, Januari 25). Forelasning 5. *PM-likströmsmotorer, Servomotorer, stegmotorer mm*. Göteborg, Göteborgs län, Sverige: Chalmers teknologiska högskola.
- Shin, P. Y. (2014, maj 20). Coupling. *AccessScience*.
- Spångberg, J., & Svensson, A. (2013). *Stegmotordriven linjärenhet i svetsmiljö*. Göteborg: Chalmers tekniska högskola.
- Trinamic. (2013, Juli 23). *PD-1160_hardware_manual.pdf*. Retrieved Maj 6, 2014, from [www.Trinamic.com: http://www.trinamic.com/products/pandrive/pandrive-stepper/pd-1160#HEAD_Downloads_Documentation](http://www.trinamic.com/products/pandrive/pandrive-stepper/pd-1160#HEAD_Downloads_Documentation)
- Trinamic. (2013, Oktober). *Appnote_stallguard2.pdf*. Retrieved Maj 6, 2014, from [www.trinamic.com: http://www.trinamic.com/products/integrated-circuits/stepper-power-driver/tmc2660](http://www.trinamic.com/products/integrated-circuits/stepper-power-driver/tmc2660)
- Trinamic. (2013, September 2). *PD-1160_TMCL_firmware_manual.pdf*. Retrieved April 29, 2014, from [www.trinamic.com: http://www.trinamic.com/products/pandrive/pandrive-stepper/pd-1160#HEAD_Downloads_Documentation](http://www.trinamic.com/products/pandrive/pandrive-stepper/pd-1160#HEAD_Downloads_Documentation)

BILAGOR

Bilaga A:

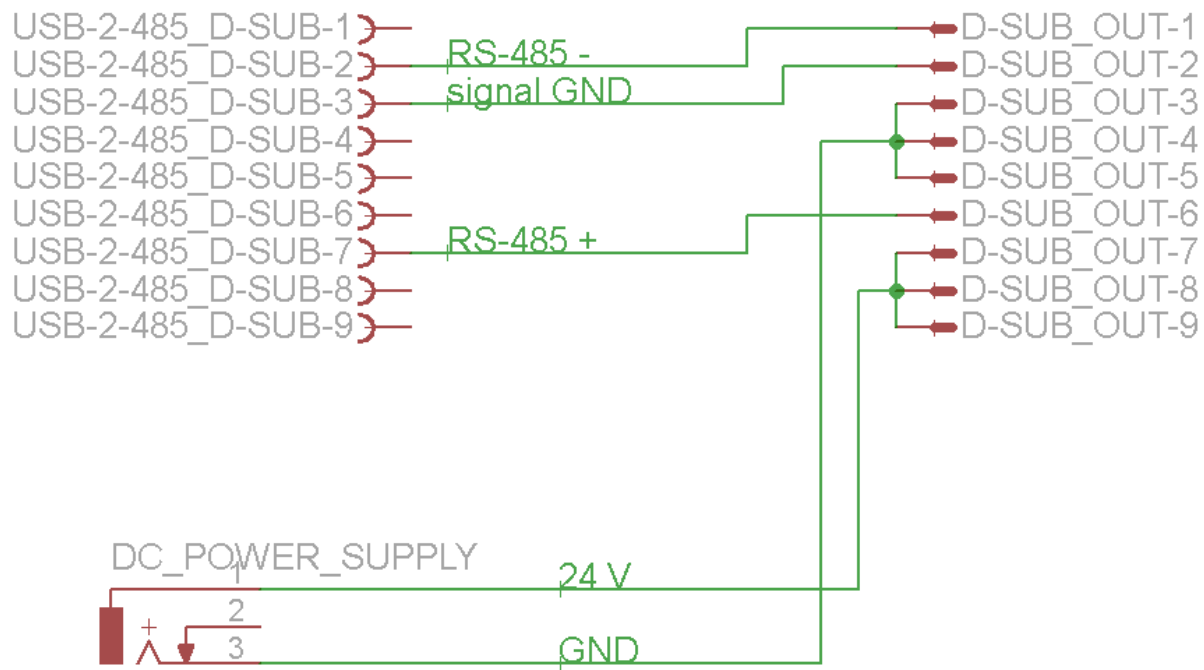
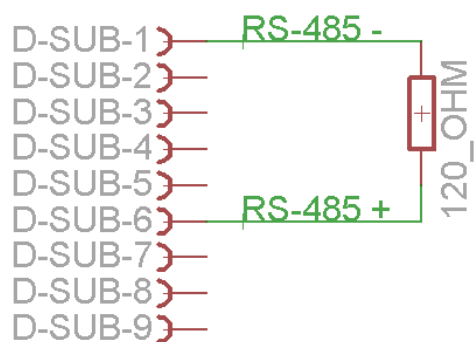
Bilaga B:

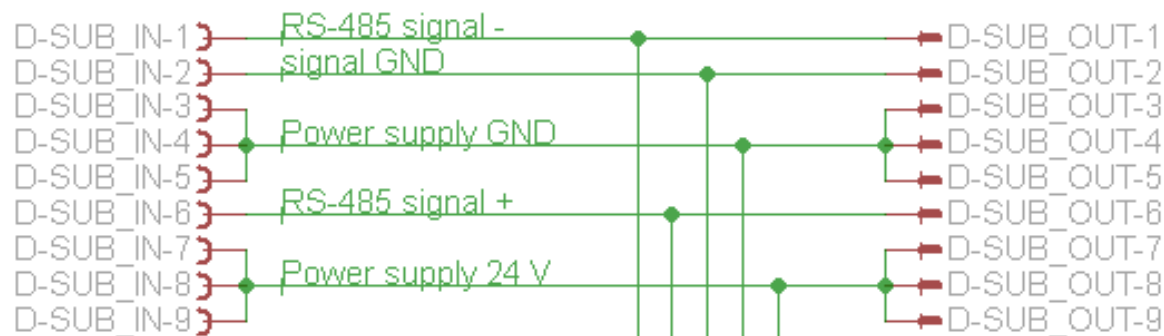
Bilaga C:

Kopplingsscheman

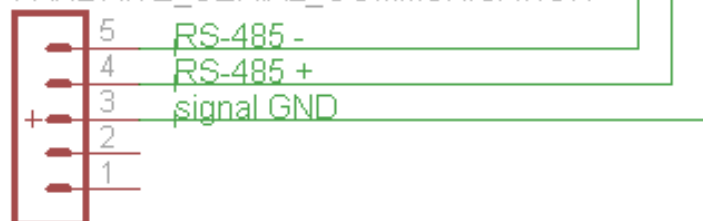
Mekaniska ritningar

Programvara

Bilaga A – 1: Centralenhet**Bilaga A – 2: Slutkontakt**



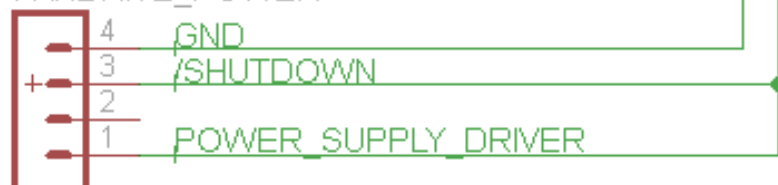
PANDRIVE_SERIAL_COMMUNICATION



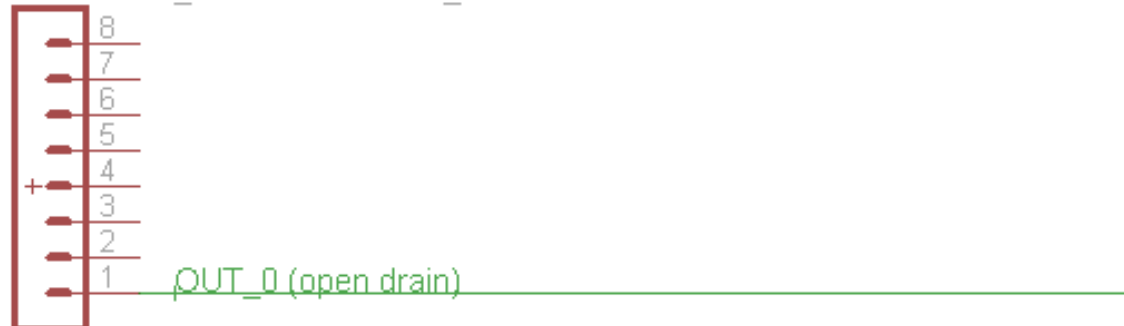
PANDRIVE_EXTERNAL_ENCODER



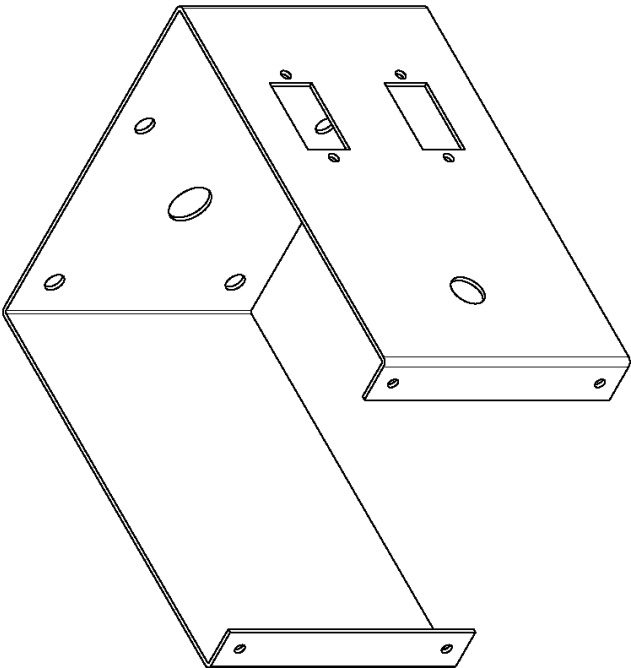
PANDRIVE_POWER



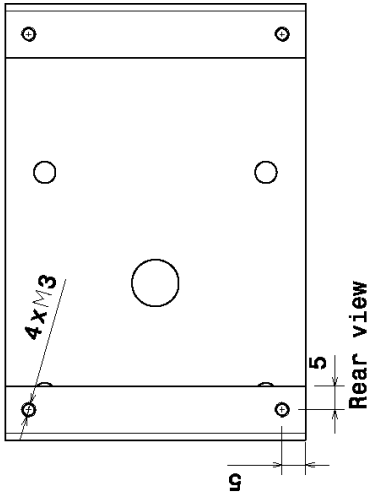
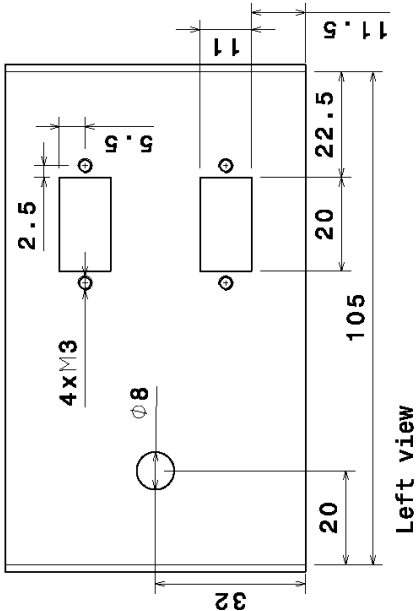
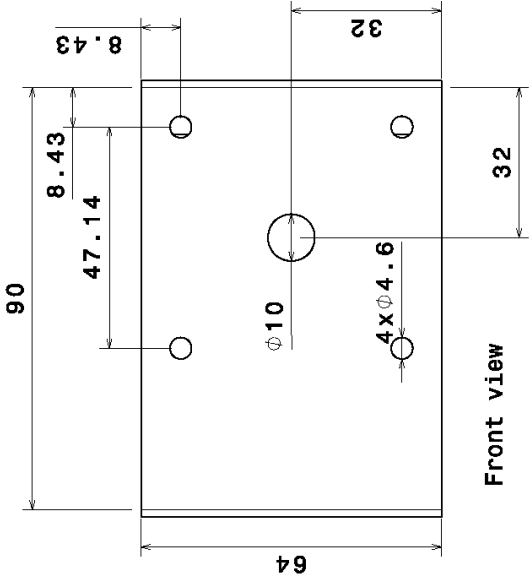
PANDRIVE_MULTIPURPOSE_I/O



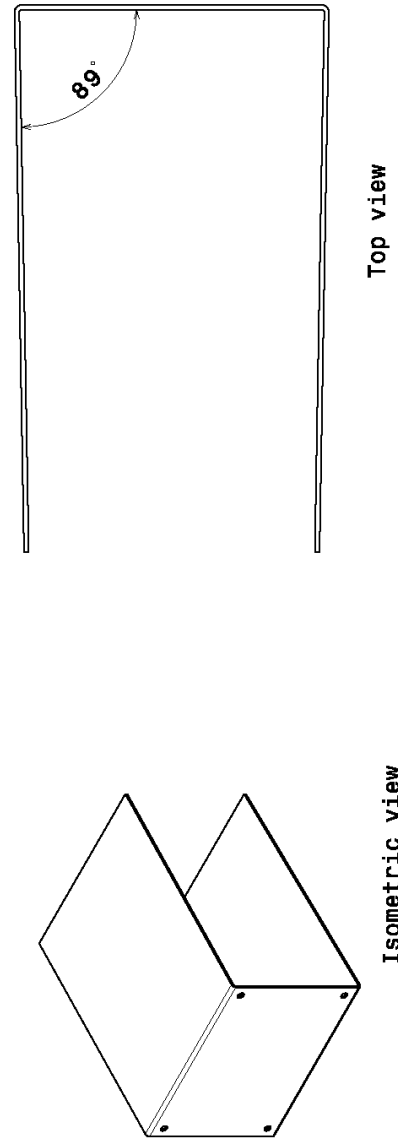
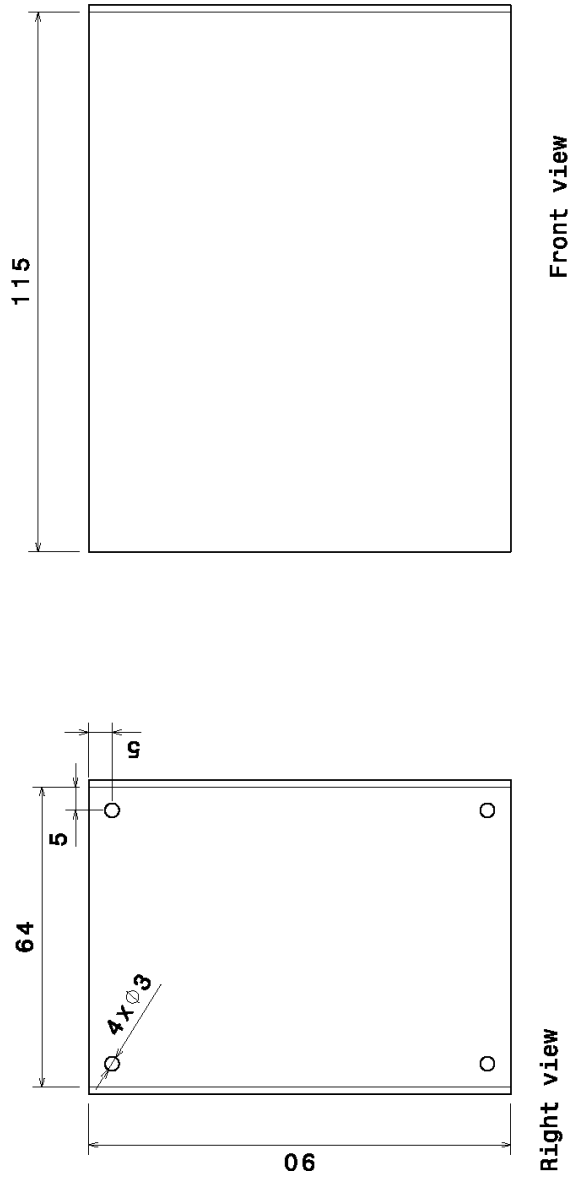
Motorlåda, del 1
Plåttjocklek 1 mm



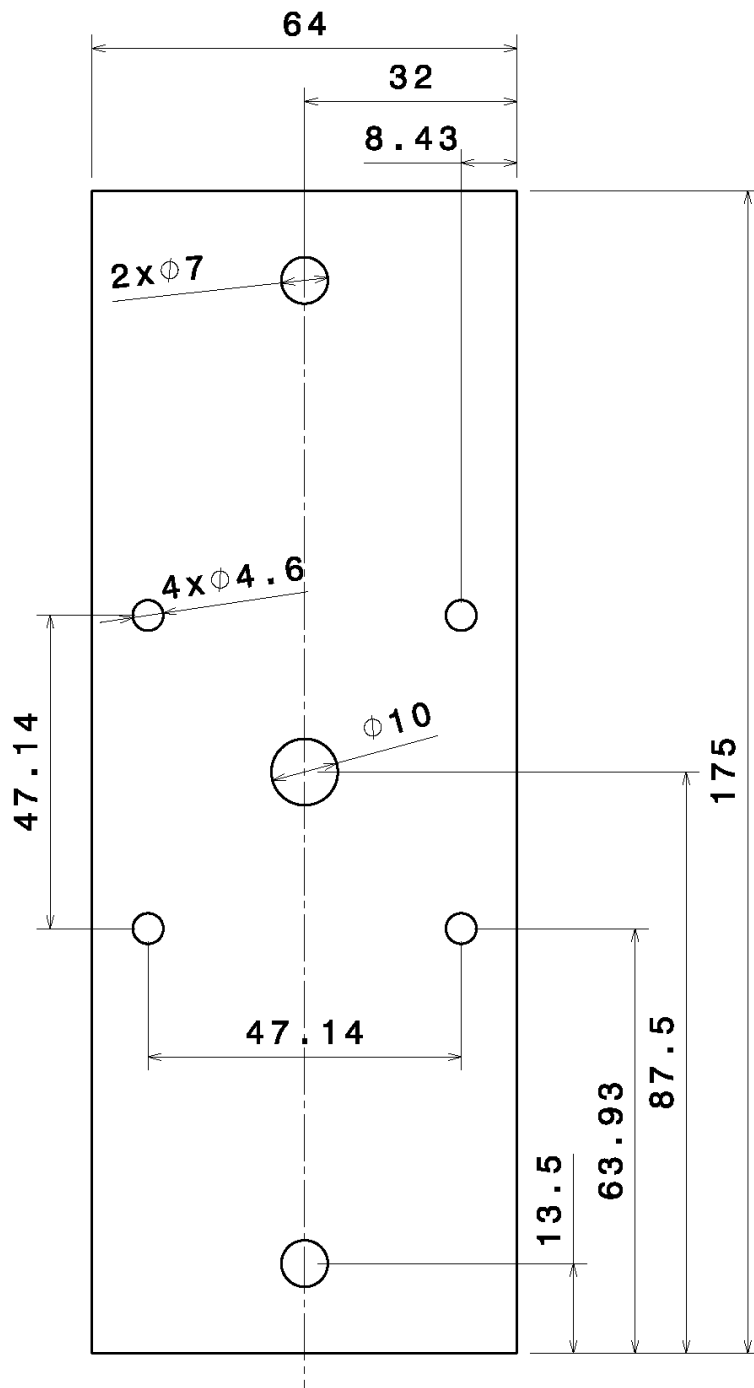
Isometric view



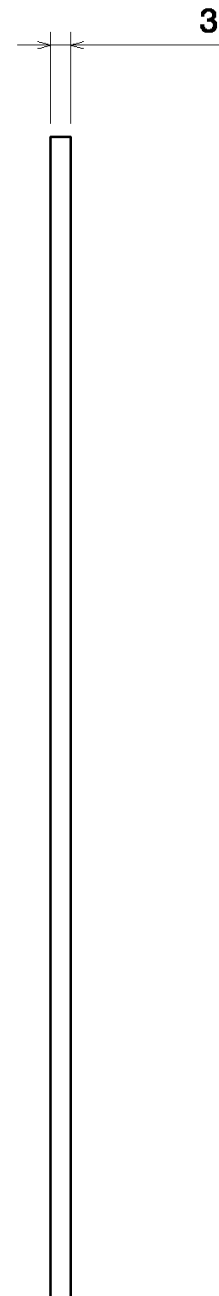
Motorlåda, del 2
Plåttjocklek 1 mm



Motorlåda, del 3



Front view



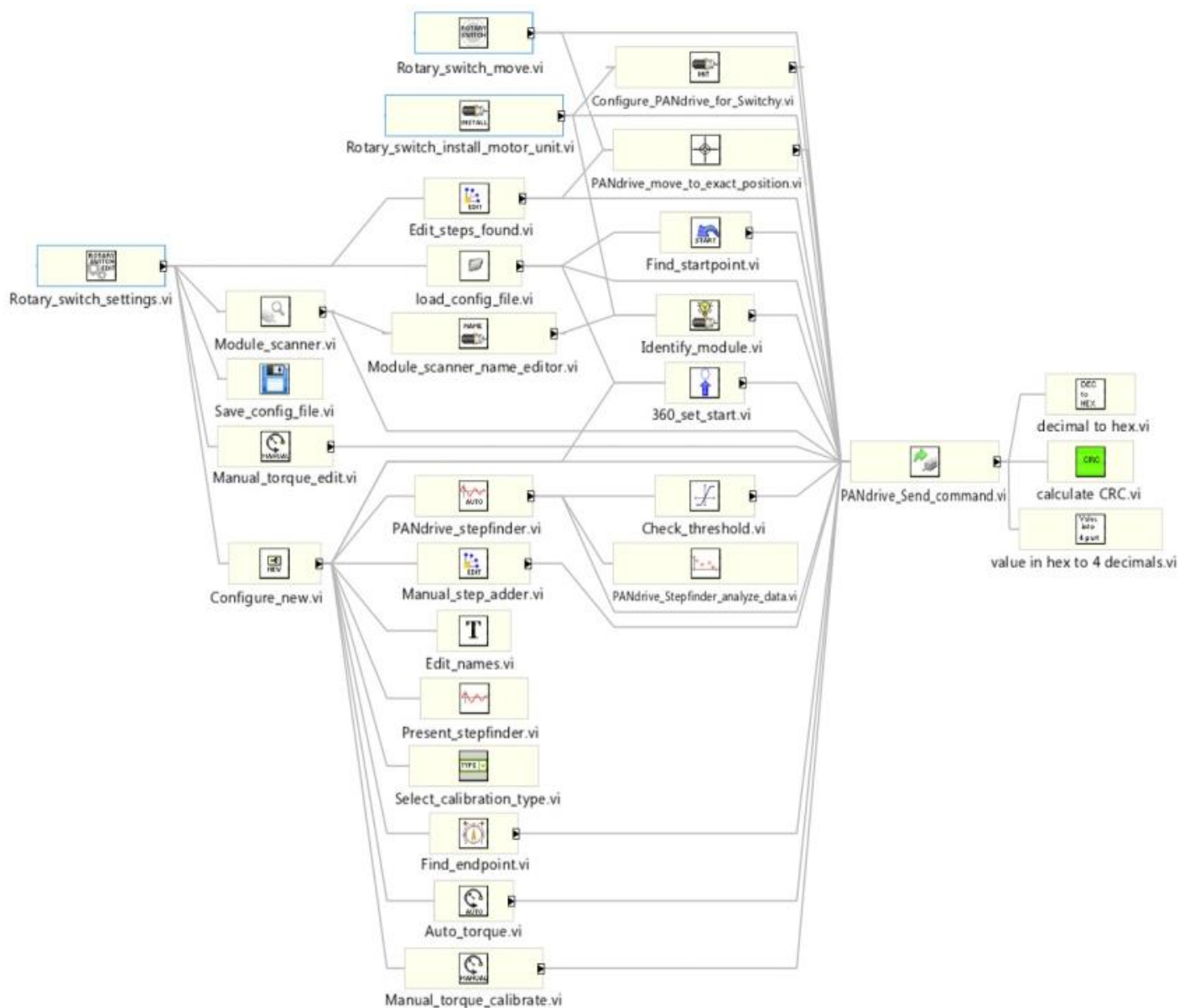
Left view

Bilaga C: Programvara

Index

VI-hierarchy.....	2
Main VI.....	3
<i>Rotary_switch_settings_VI-version.vi</i>	<i>3</i>
<i>Rotary_switch_settings.exe</i>	<i>8</i>
<i>Rotary_switch_move_VI-version.vi</i>	<i>14</i>
<i>Rotary_switch_move.exe</i>	<i>17</i>
<i>Rotary_switch_install_motor_unit.exe</i>	<i>21</i>
Sub VI.....	26
<i>360_set_start.vi</i>	<i>26</i>
<i>Configure_PANdrive_for_Switchy.vi</i>	<i>28</i>
<i>Auto_torque.vi</i>	<i>31</i>
<i>Calculate_CRC.vi</i>	<i>34</i>
<i>Check_threshold.vi</i>	<i>35</i>
<i>Configure_new.vi</i>	<i>37</i>
<i>Decimal to hex.vi</i>	<i>40</i>
<i>Edit_names.vi</i>	<i>41</i>
<i>Edit_steps_found.vi</i>	<i>46</i>
<i>Find_endpoint.vi</i>	<i>56</i>
<i>Find_startpoint.vi</i>	<i>58</i>
<i>Identify_module.vi</i>	<i>59</i>
<i>load_config_file.vi</i>	<i>60</i>
<i>Manual_step_adder.vi</i>	<i>64</i>
<i>Manual_torque_calibrate.vi</i>	<i>72</i>
<i>Manual_torque_edit.vi</i>	<i>74</i>
<i>Module_scanner.vi</i>	<i>76</i>
<i>Module_scanner_name_editor.vi</i>	<i>79</i>
<i>PANdrive_move_to_exact_position.vi</i>	<i>82</i>
<i>PANdrive_Send_command.vi</i>	<i>83</i>
<i>PANdrive_stepfinder.vi</i>	<i>89</i>
<i>PANdrive_Stepfinder_analyze_data.vi</i>	<i>95</i>
<i>Present_stepfinder.vi</i>	<i>100</i>
<i>Save_config_file.vi</i>	<i>102</i>
<i>Select_calibration_type.vi</i>	<i>103</i>
<i>Value in hex to 4 decimals.vi</i>	<i>104</i>

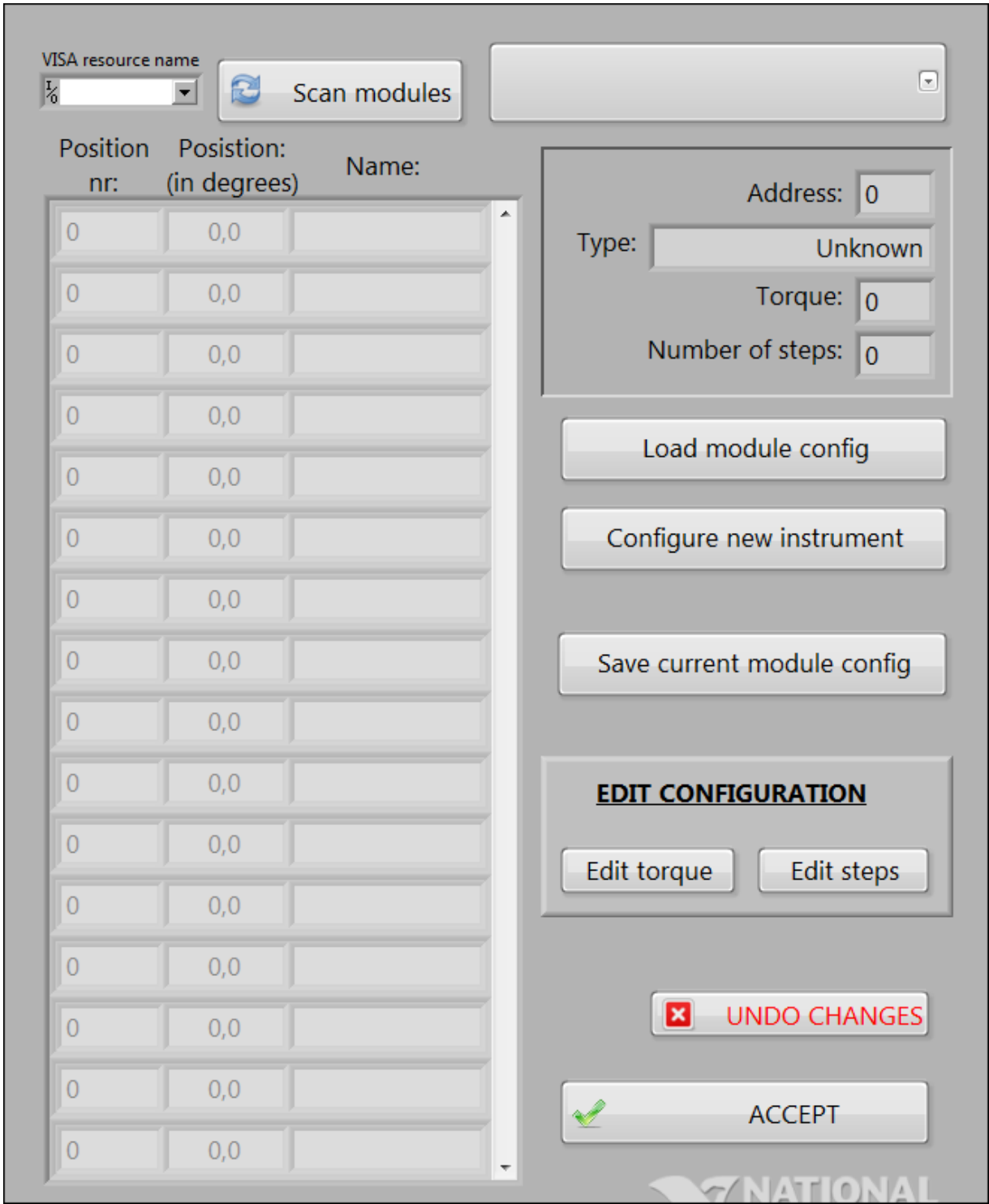
VI-hierarchy



Main VI

Rotary_switch_settings_VI-version.vi

Rotary_switch calibration editor. Used to configure rotary switches and give them ID-names, and also find and name their respective step positions.



VISA resource name:

 Scan modules

Position nr:	Position: (in degrees)	Name:
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	

Address:

Type:

Torque:

Number of steps:

Load module config

Configure new instrument

Save current module config

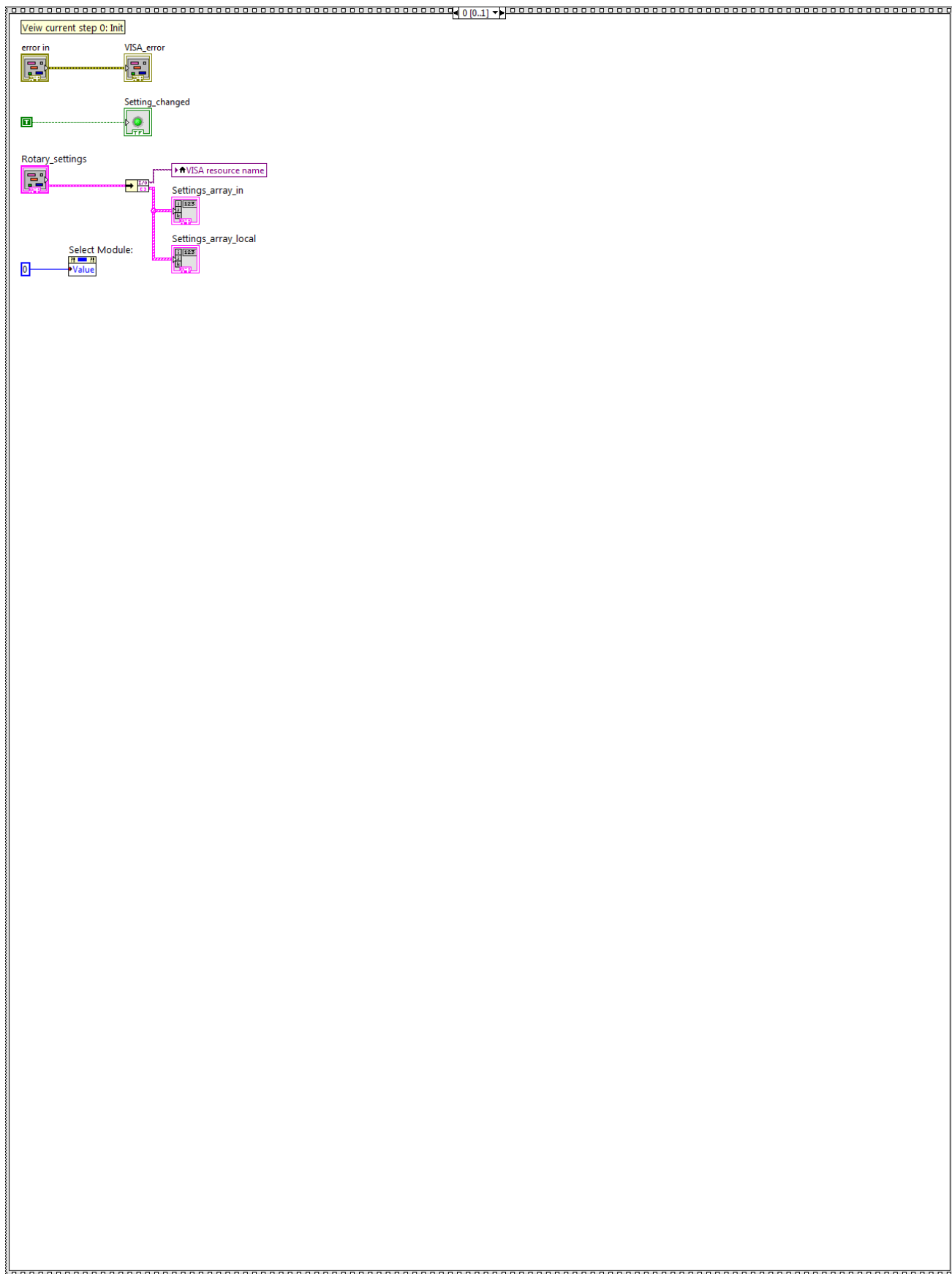
EDIT CONFIGURATION

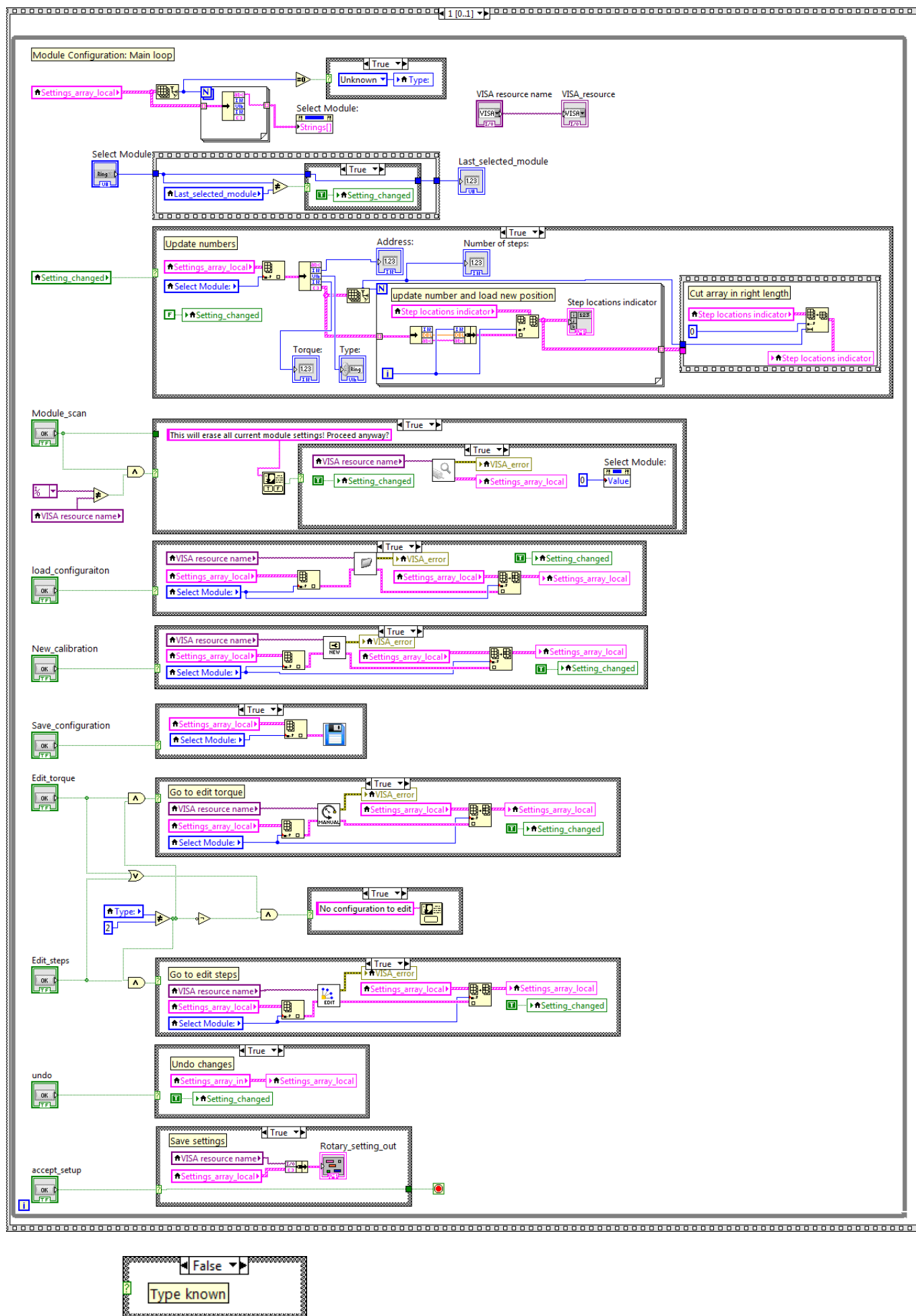
Edit torque Edit steps

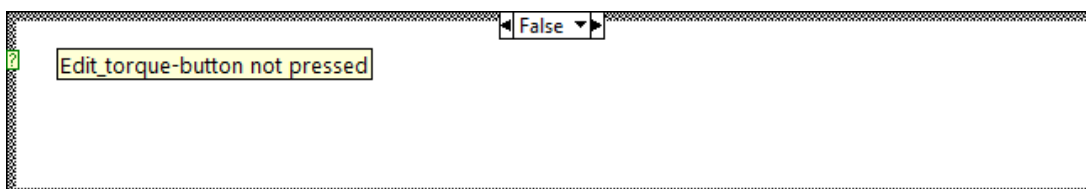
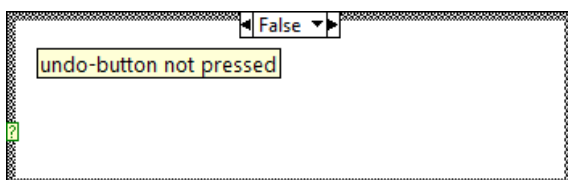
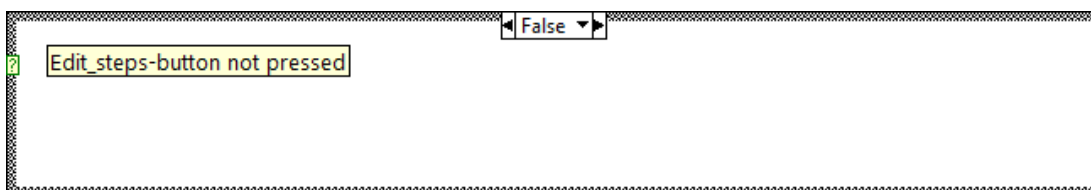
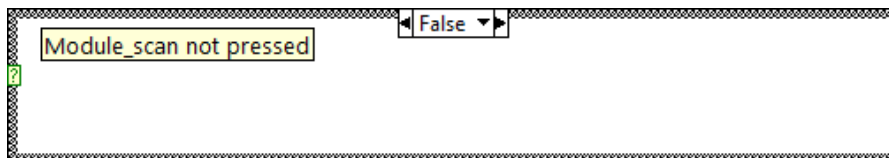
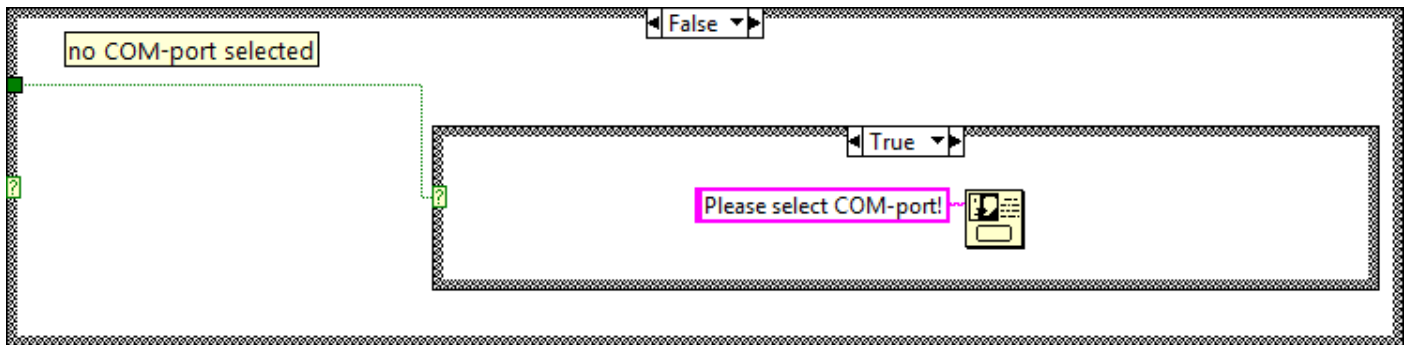
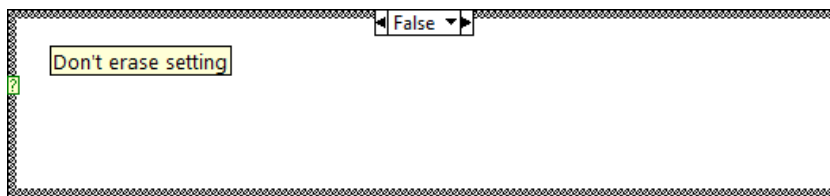
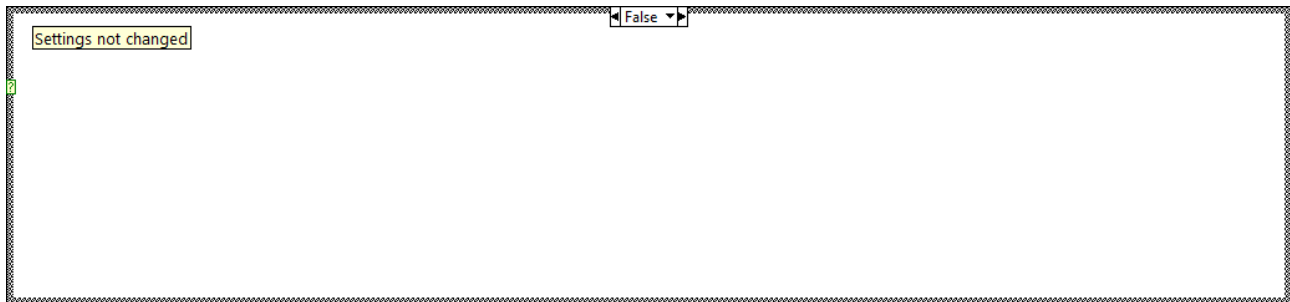
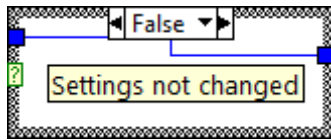
 UNDO CHANGES

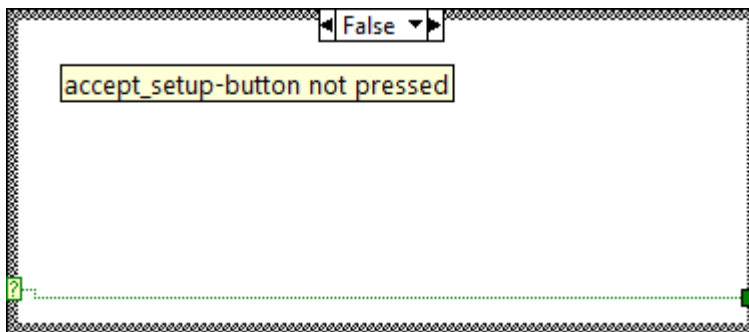
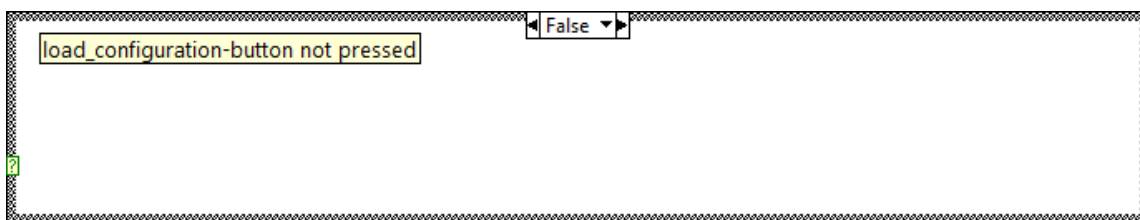
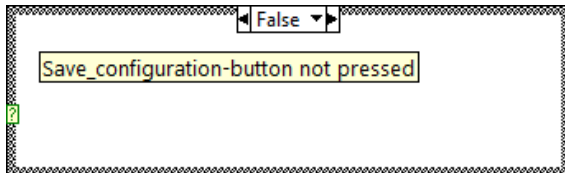
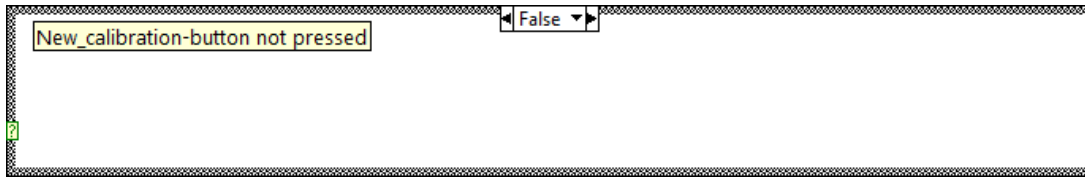
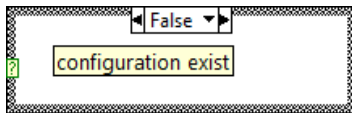
 ACCEPT











Rotary_switch_settings.exe

Rotary_switch calibration editor. Used to configure rotary switches and give them ID-names, and also find and name their respective step positions.

This is the executable file-version which saves calibration into a file!

The screenshot shows the Rotary_switch_settings.exe calibration editor interface. It features a table for configuring rotary switches, a 'Scan modules' button, and various configuration options on the right.

VISA resource name: Scan modules

Position nr:	Position: (in degrees)	Name:
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	

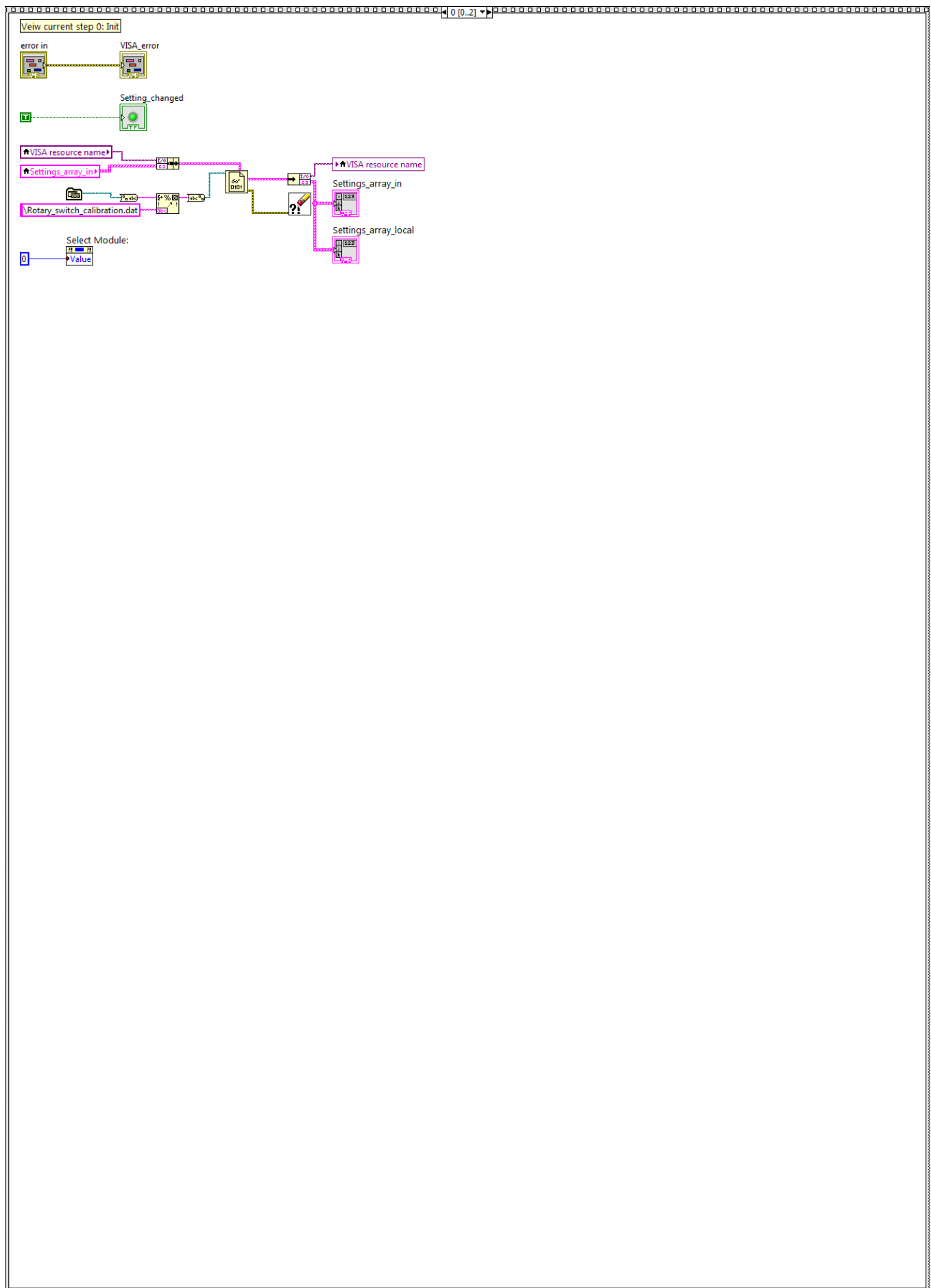
Configuration options:

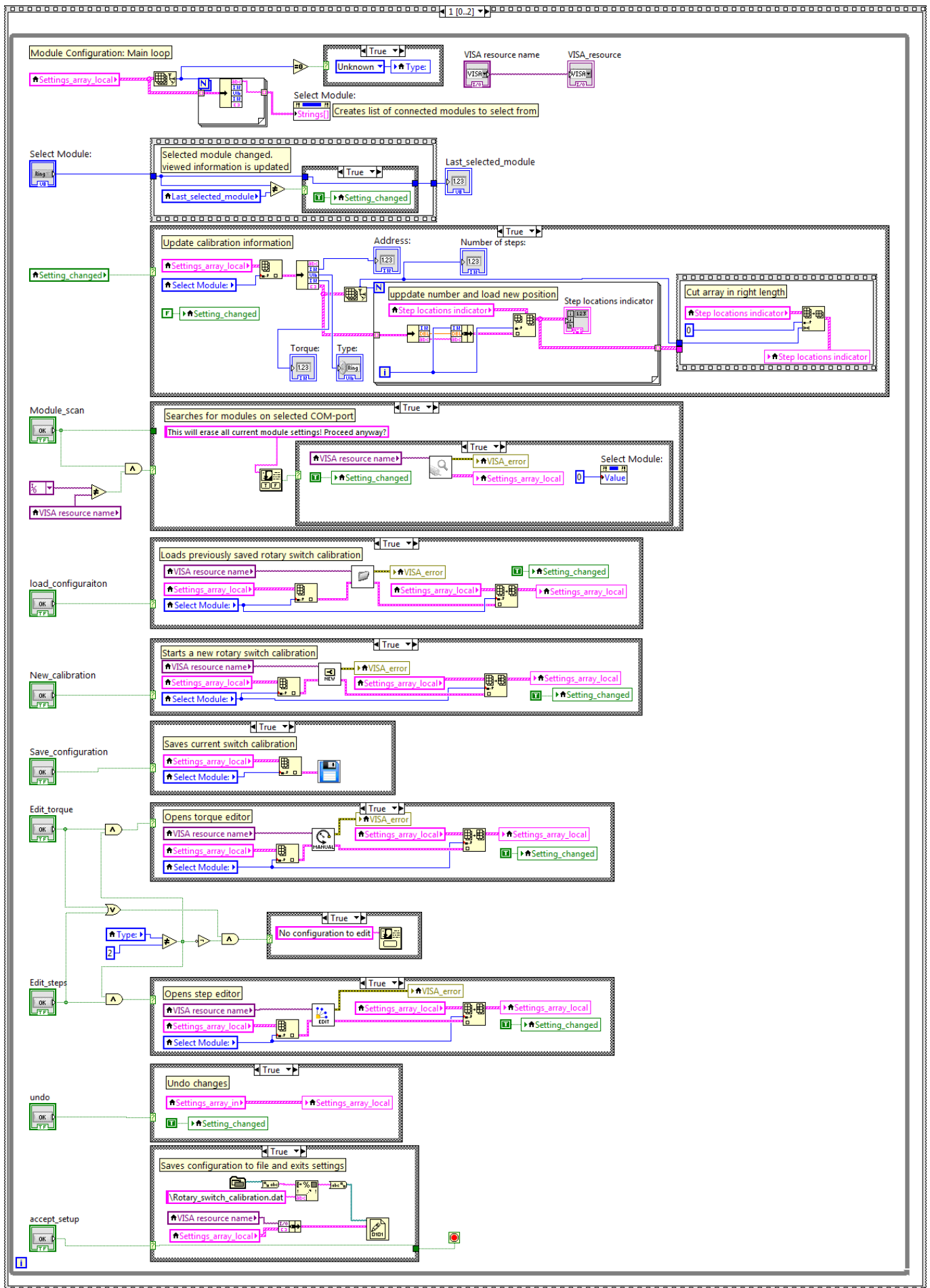
- Address:
- Type:
- Torque:
- Number of steps:

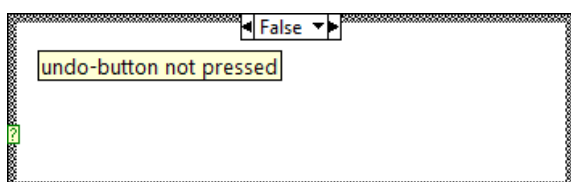
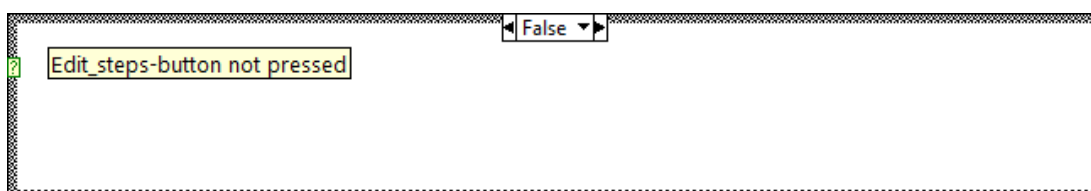
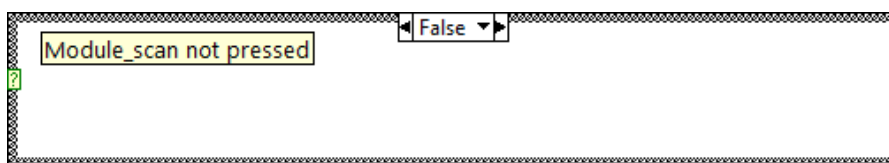
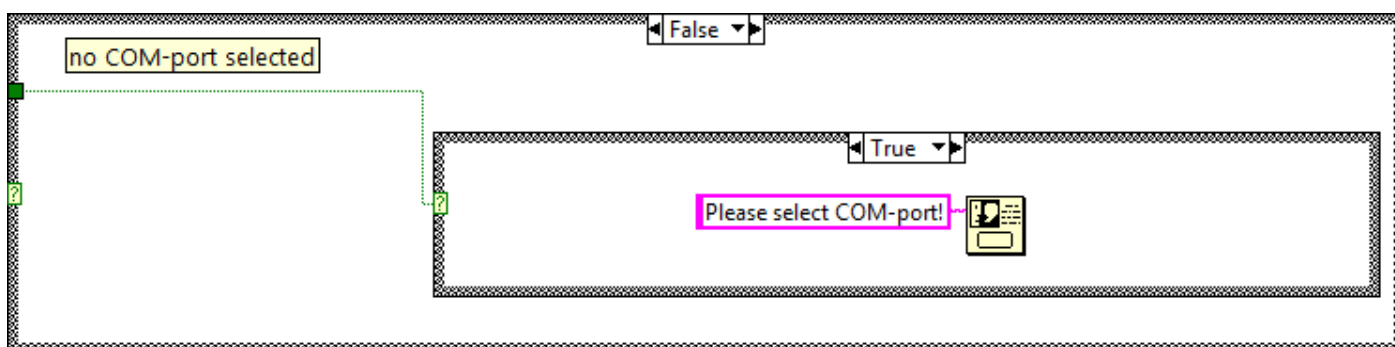
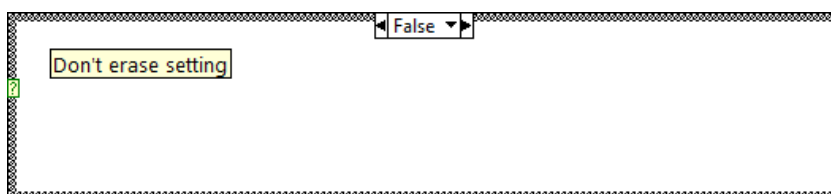
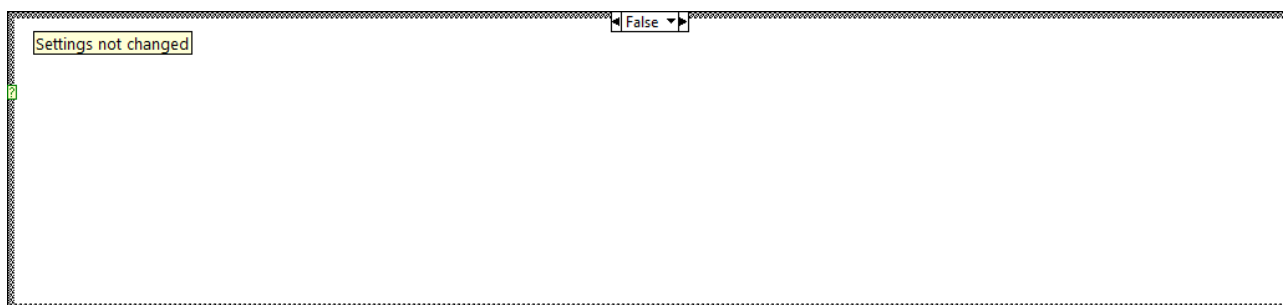
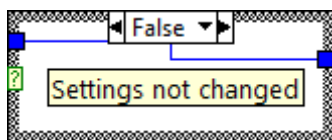
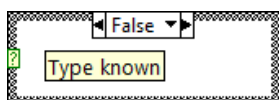
Buttons:

- Load module config
- Configure new instrument
- Save current module config
- EDIT CONFIGURATION**
- Edit torque
- Edit steps
- UNDO CHANGES
- ACCEPT

Logo: NATIONAL







False

Edit_torque-button not pressed

False

configuration exist

False

New_calibration-button not pressed

False

Save_configuration-button not pressed

False

load_configuration-button not pressed

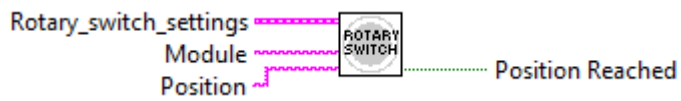
False

accept_setup-button not pressed

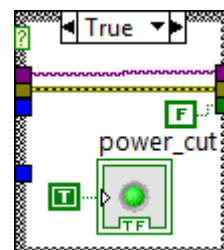
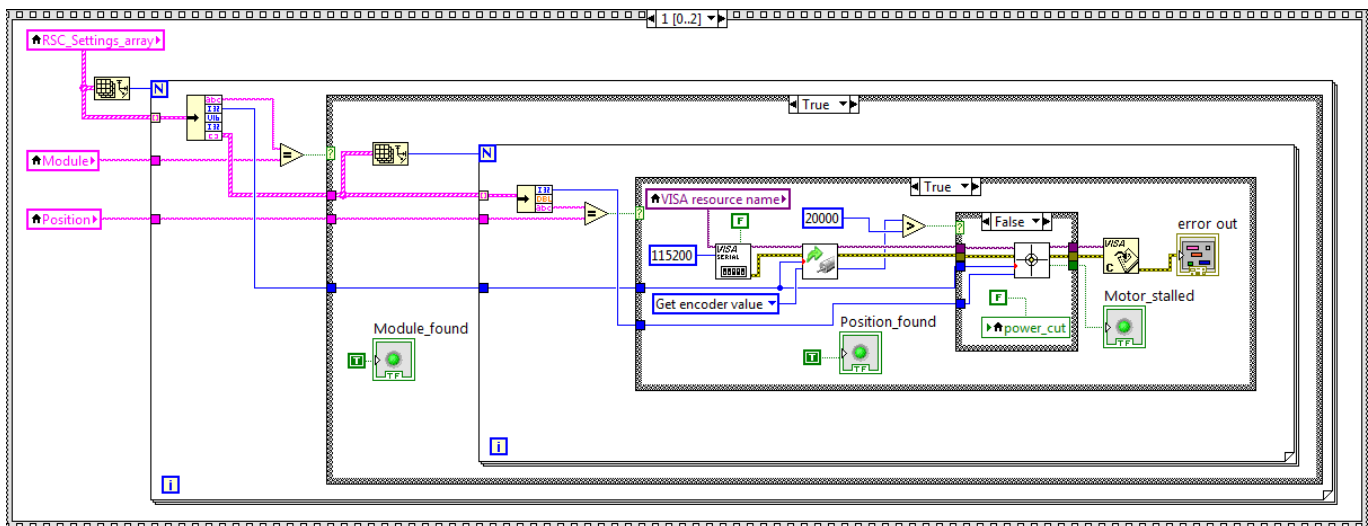
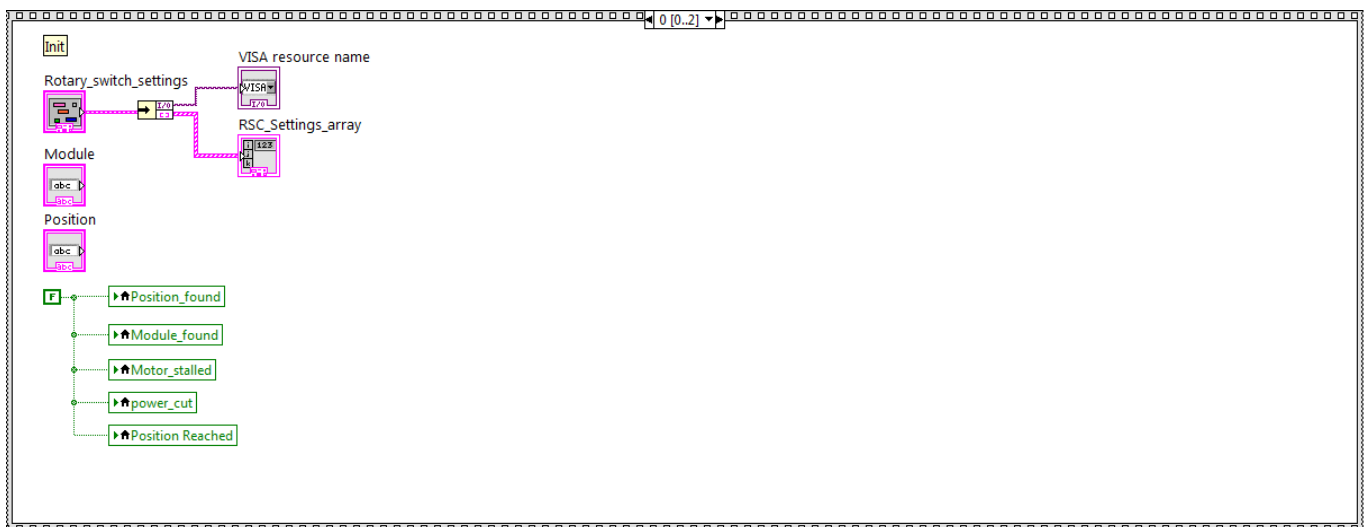


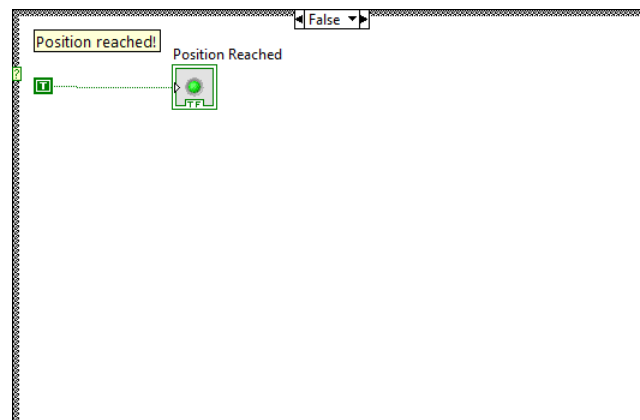
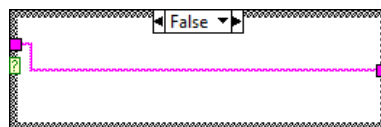
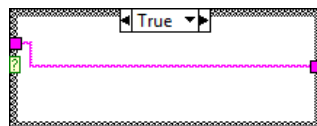
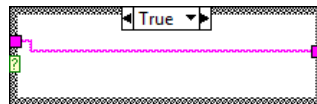
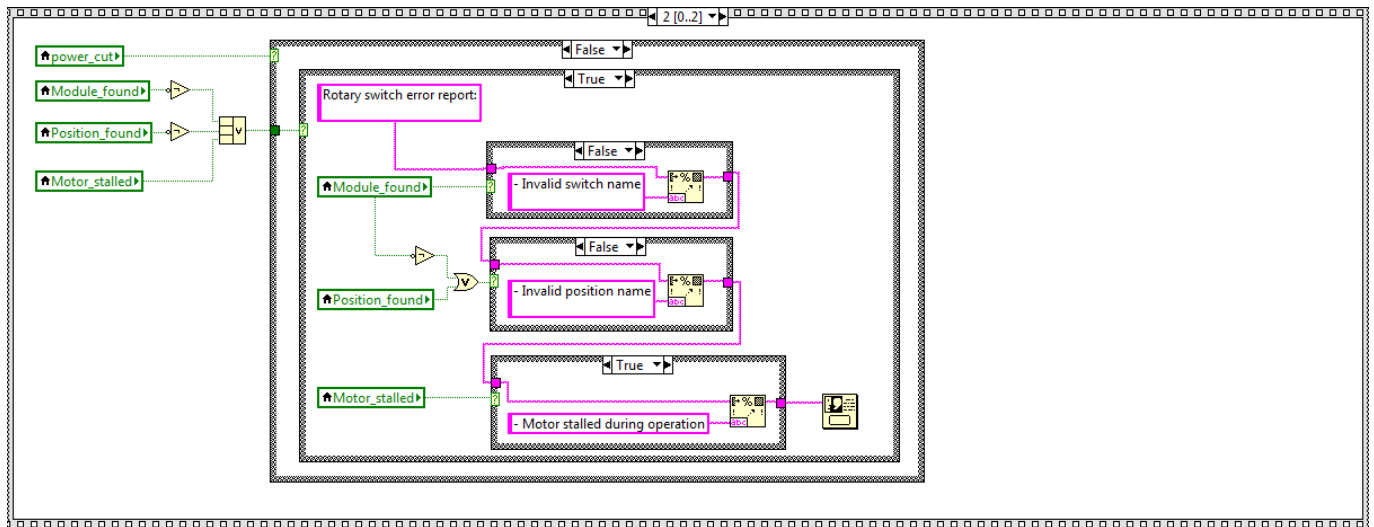
Rotary_switch_move_VI-version.vi

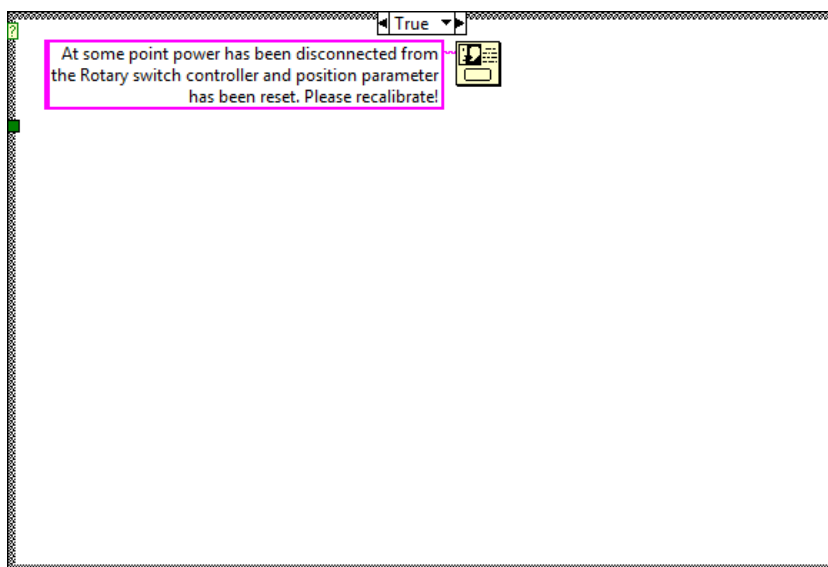
Moves the switch to a chosen position.



Operating rotary switch...





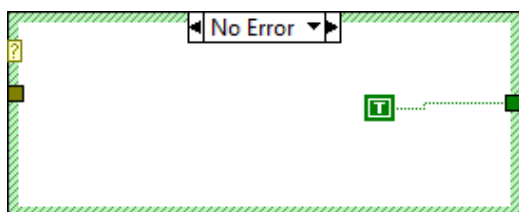
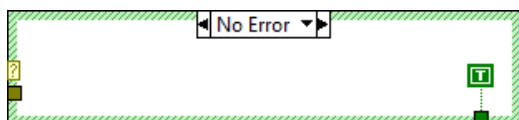
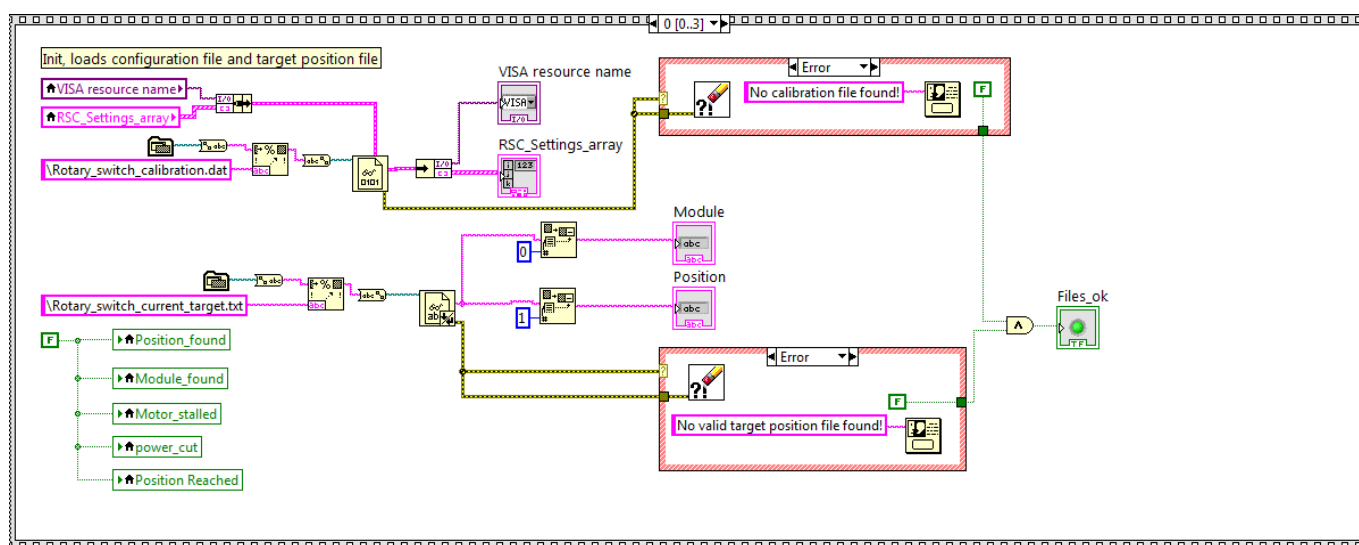


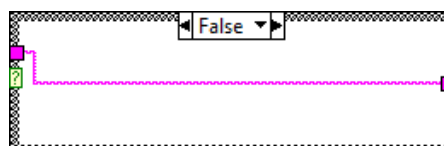
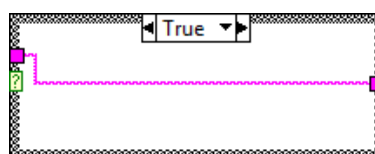
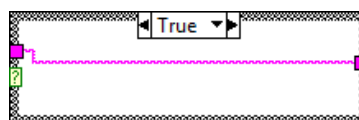
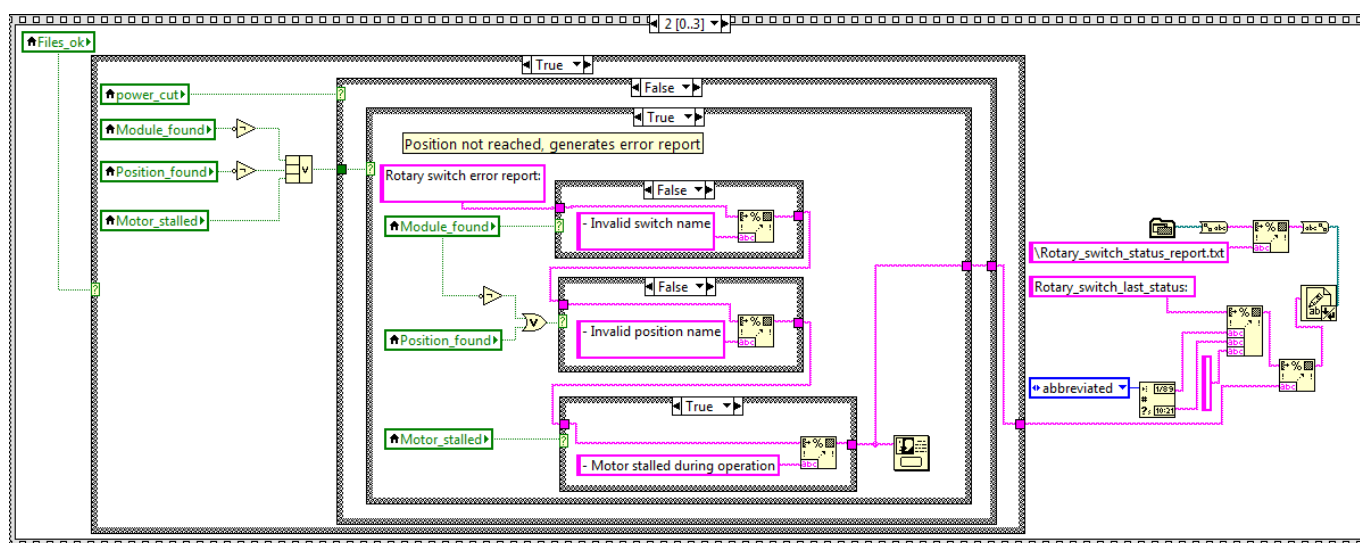
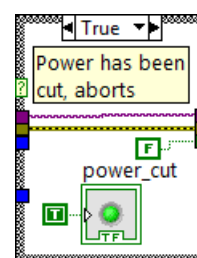
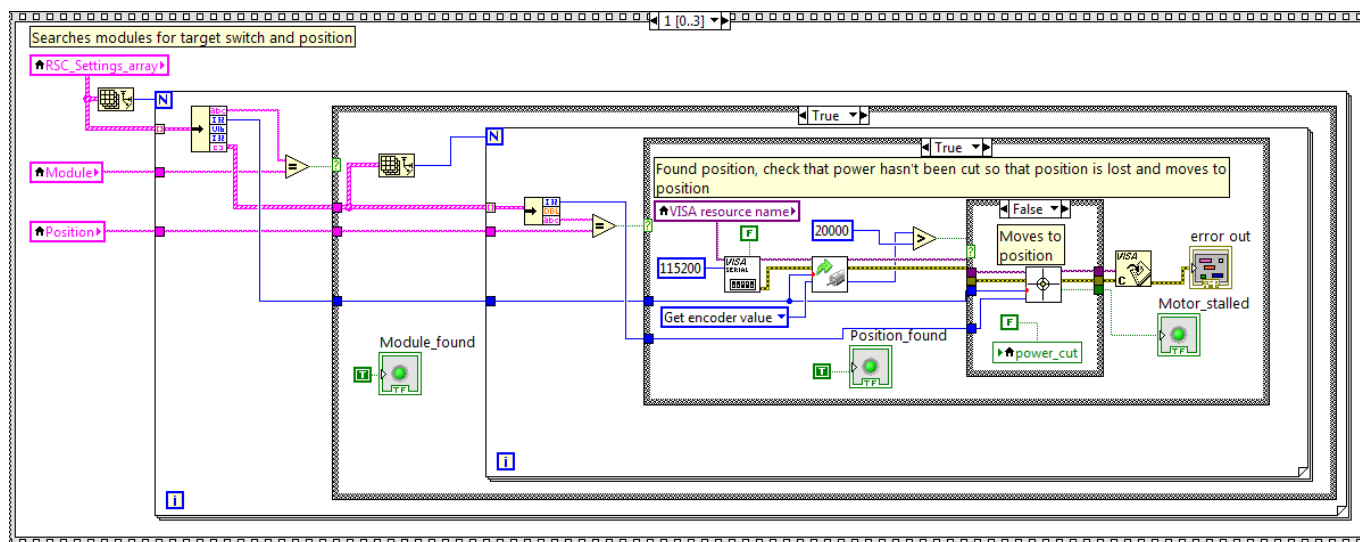
Rotary_switch_move.exe

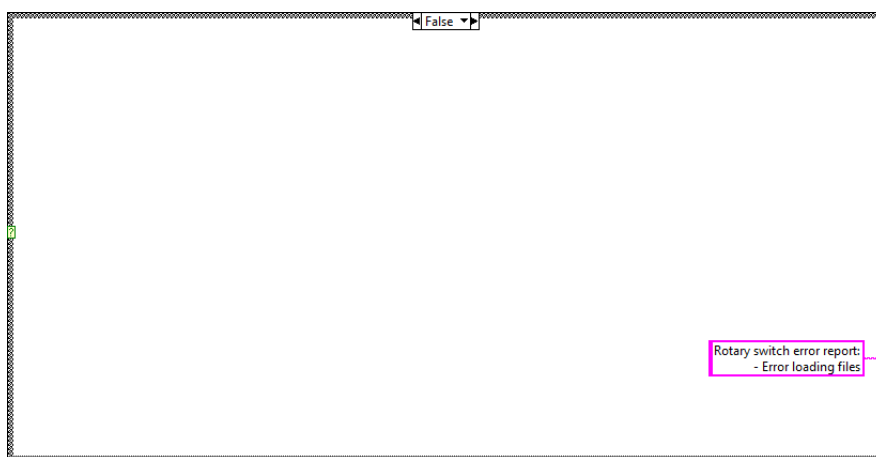
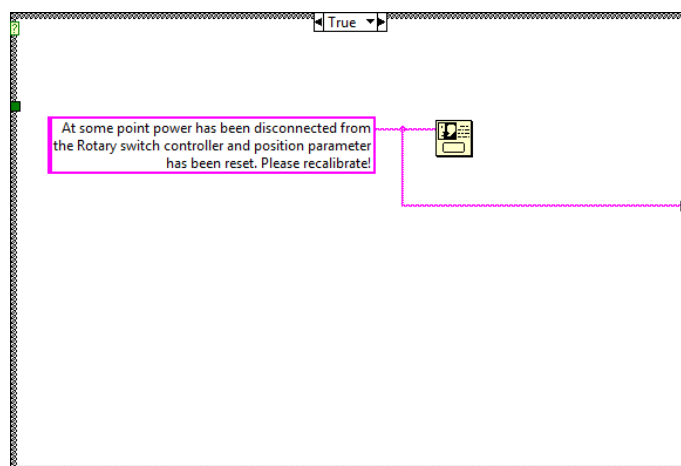
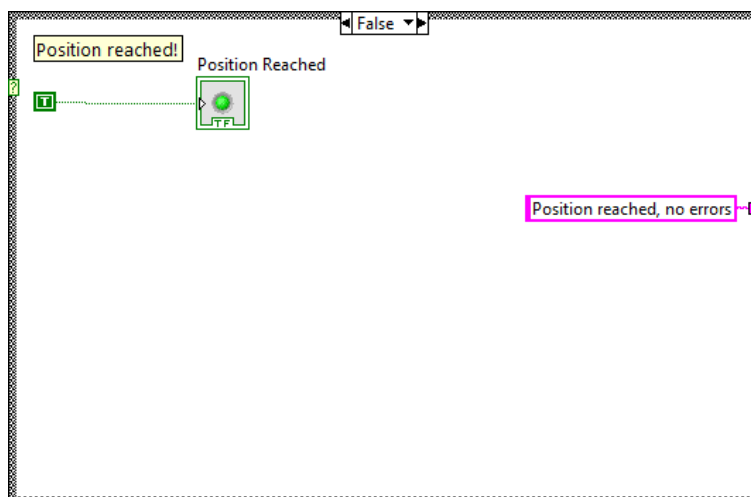
Moves the switch to a chosen position.

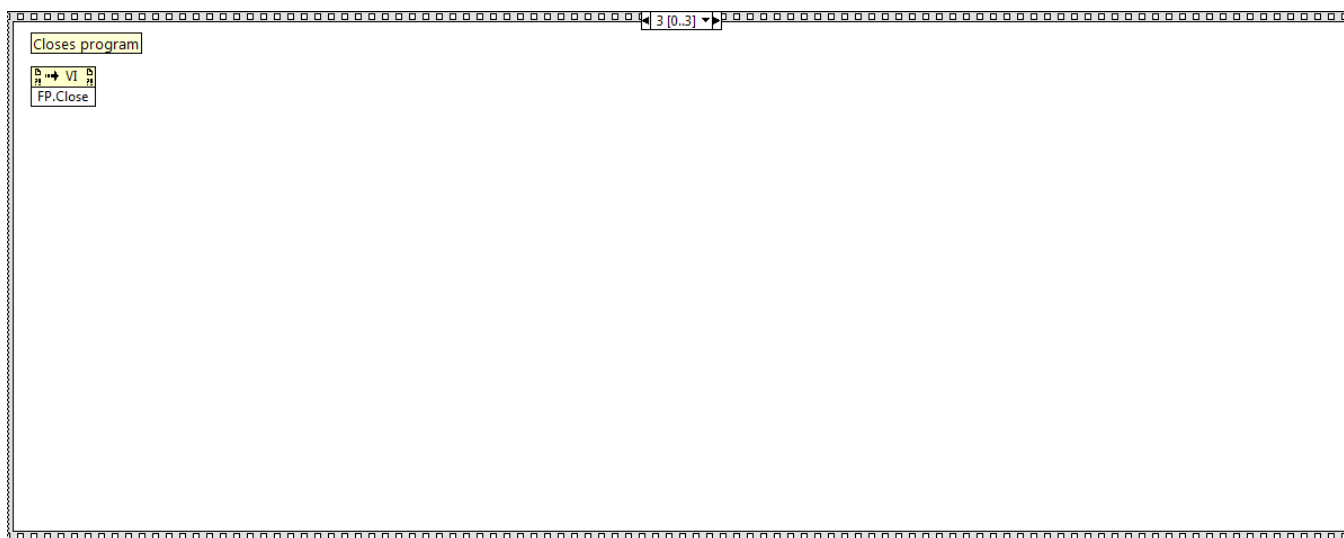
This is the executable file-version that gets its target position from a textfile and loads a calibration-file generated by "Rotary_switch_settings.exe"!

Operating rotary switch...









Rotary_switch_install_motor_unit.exe

Loads PANdrive with a small program to make it compatible with the Rotary_switch-system. Lets the user select which module address the configured module will have.

This is compiled as an executable file only, it is not built for interacting with other VI:s.

The application window has a light gray background and contains two main sections: "SCAN MODULES" and "CONFIGURE MODULE".

SCAN MODULES

Select baud rates to scan:

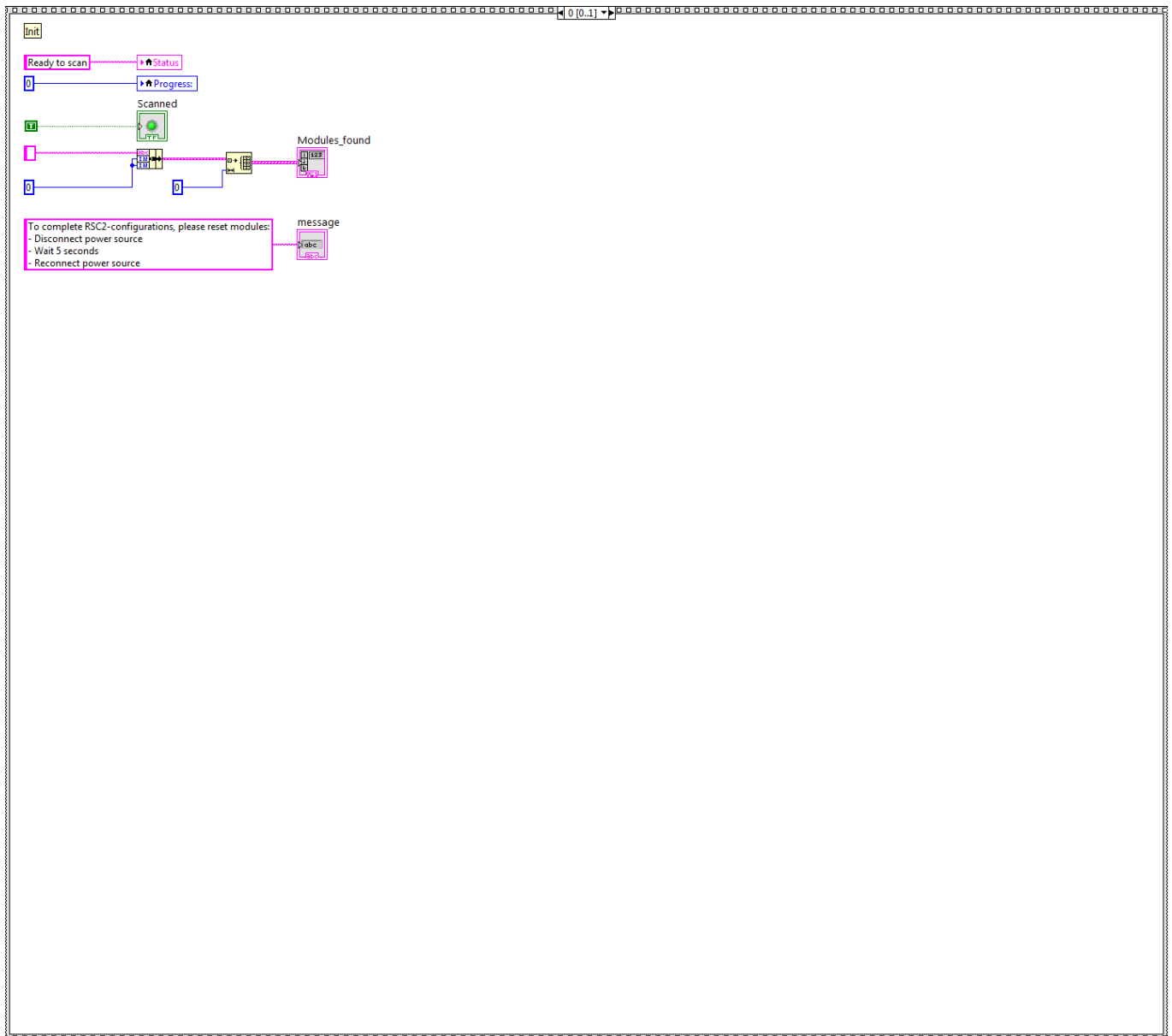
Standard PANdrive

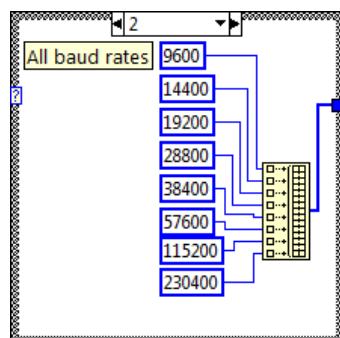
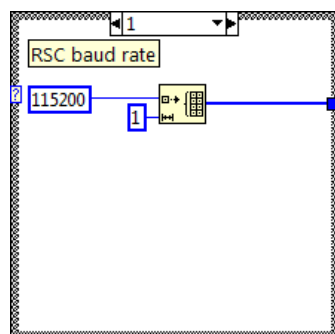
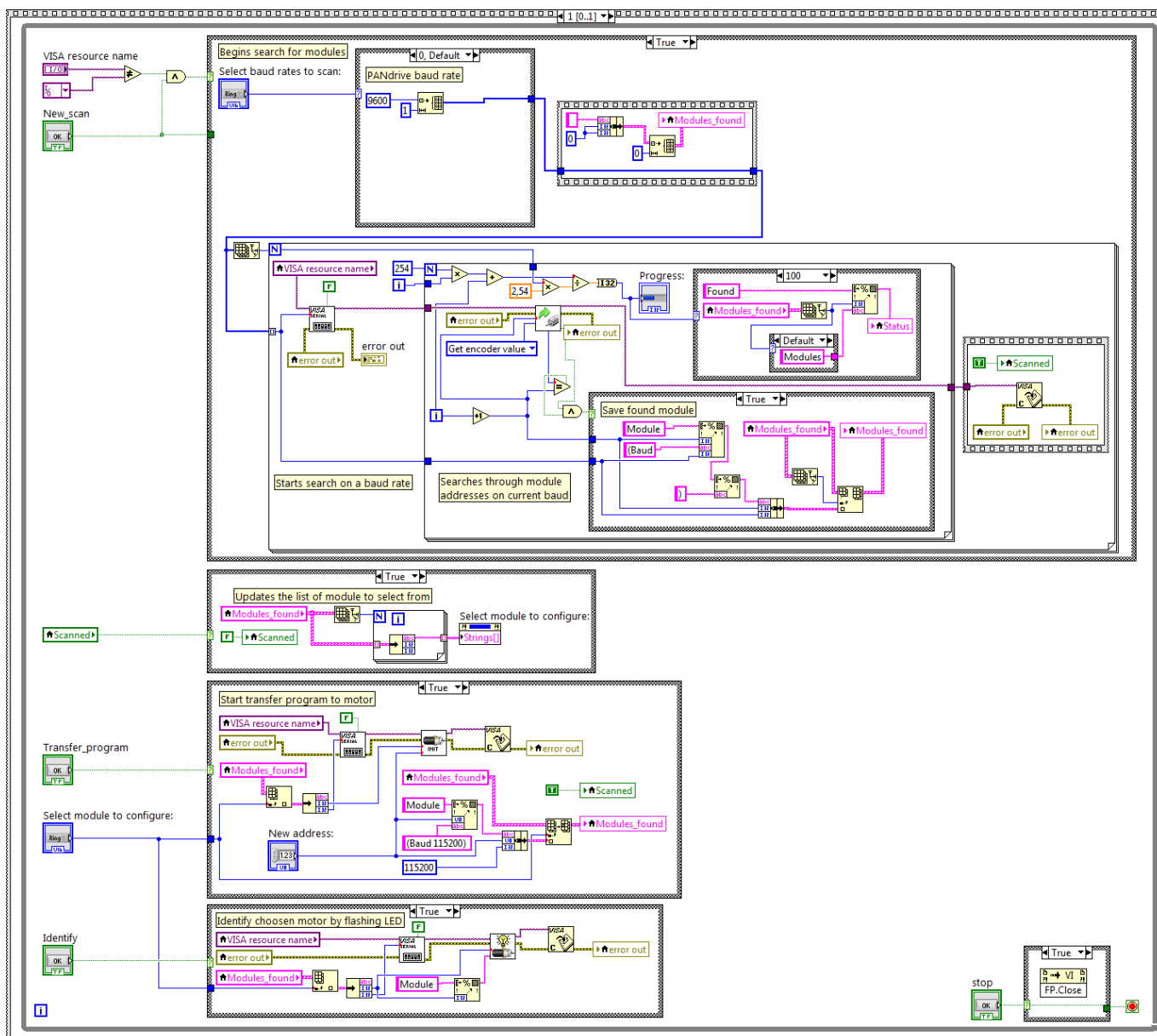
CONFIGURE MODULE

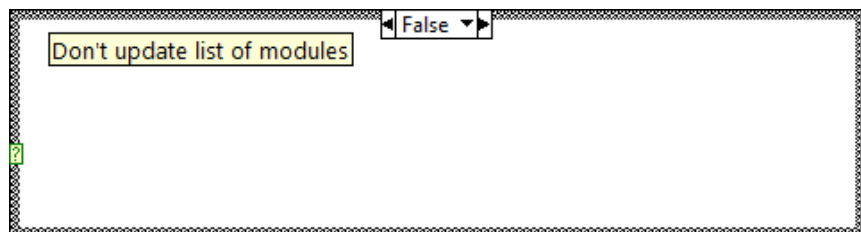
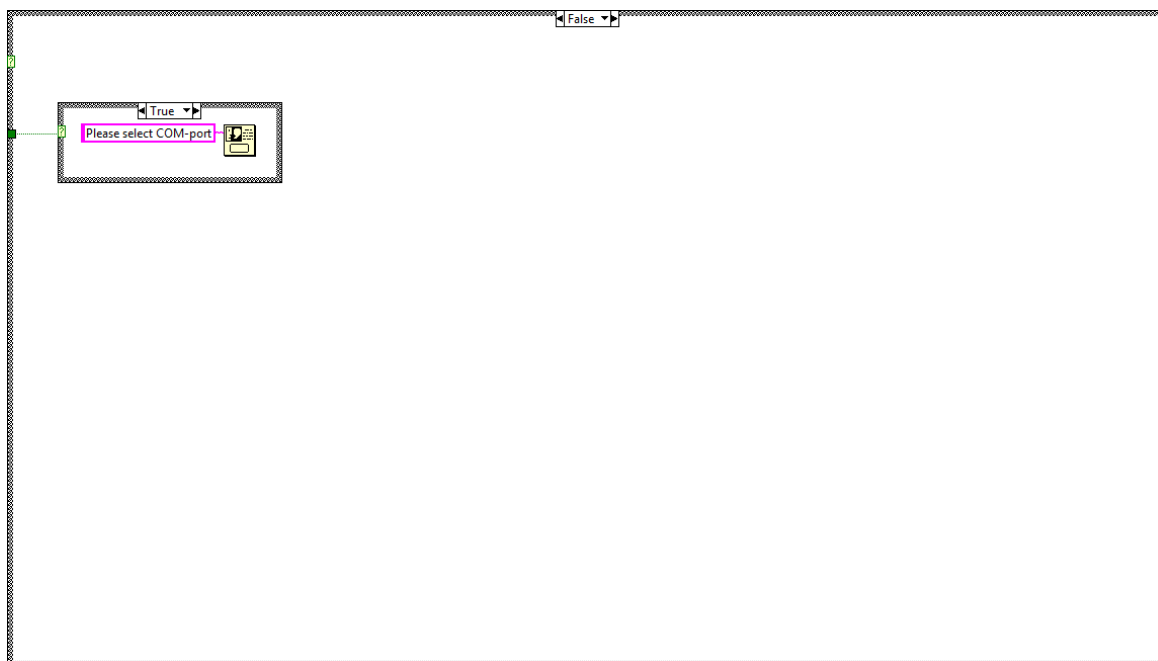
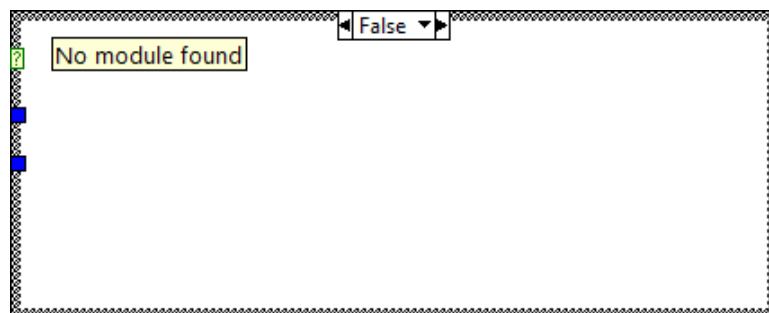
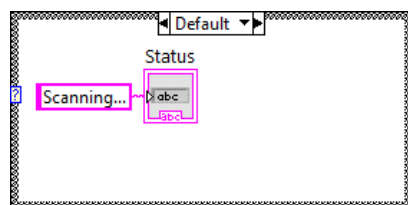
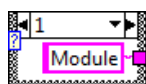
Select module to configure:

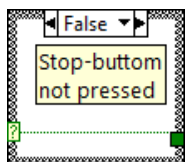
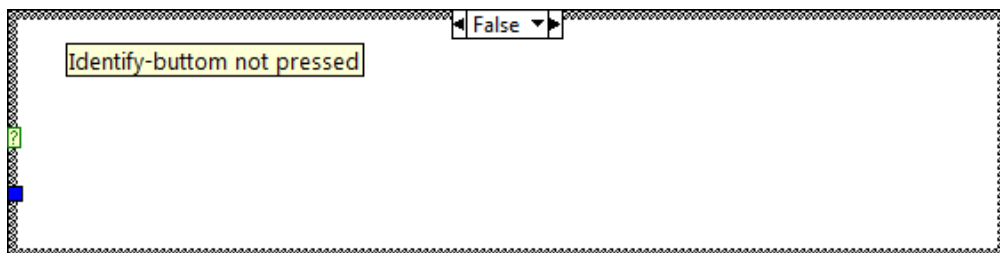
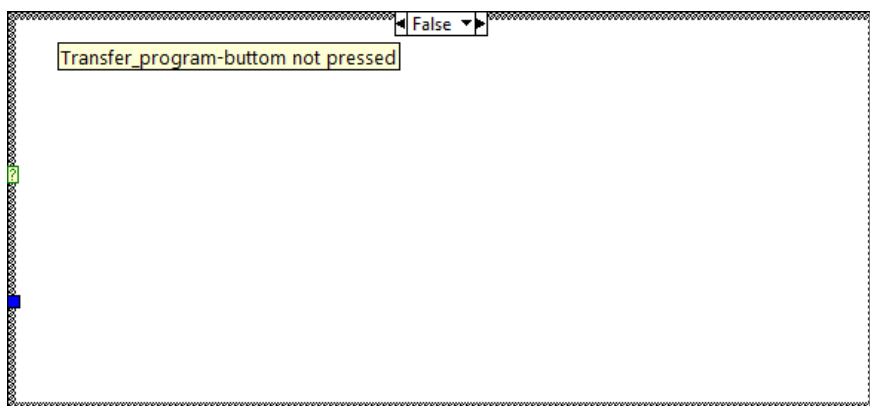
Module 1 (Baud 115200)

New address:





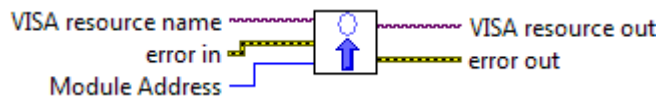




Sub VI

360_set_start.vi

Sets the start point (zero degree position) for a 360 degree switch

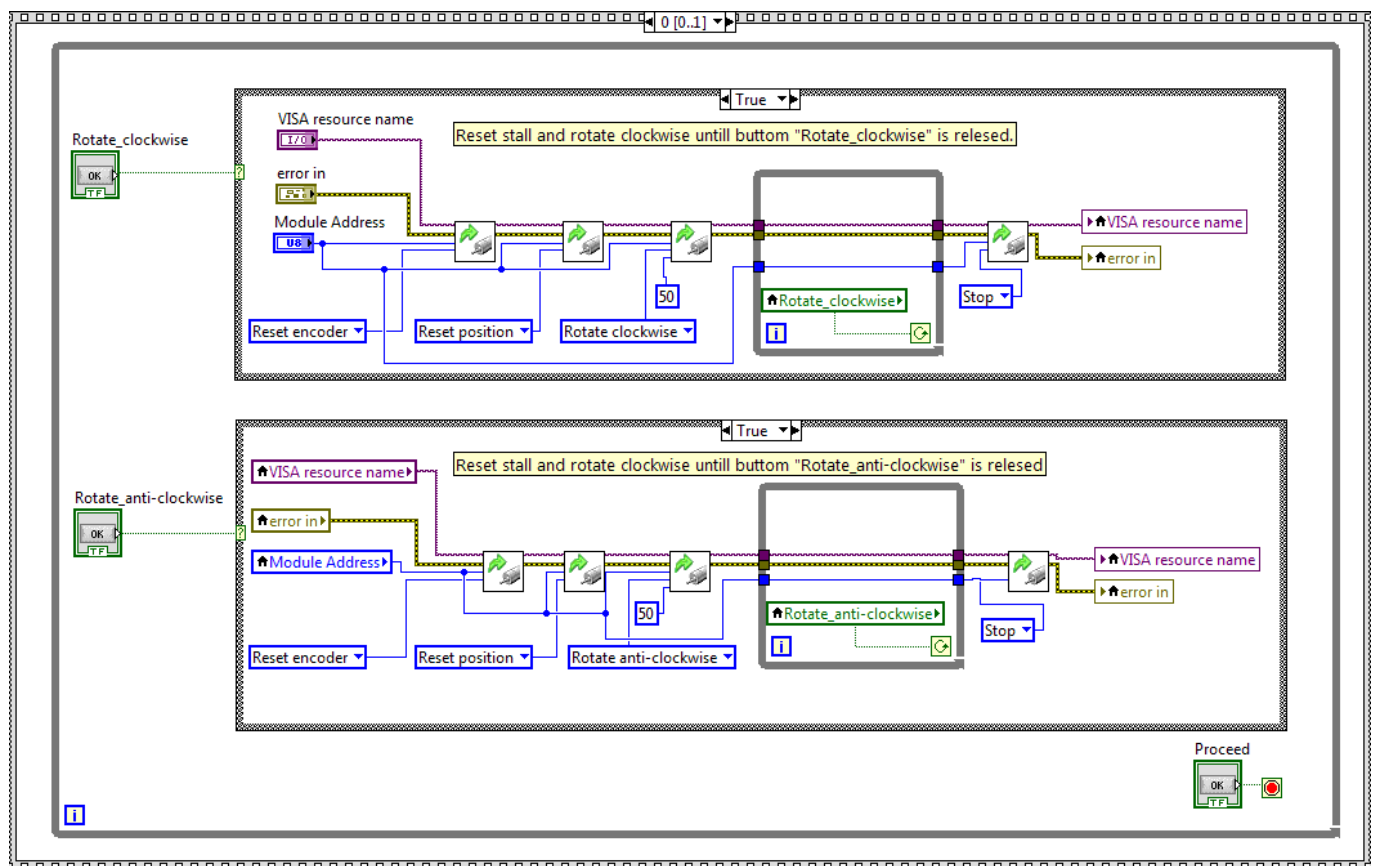


Set the switch to its start position

✓ DONE

Rotate clockwise

Rotate anti-clockwise

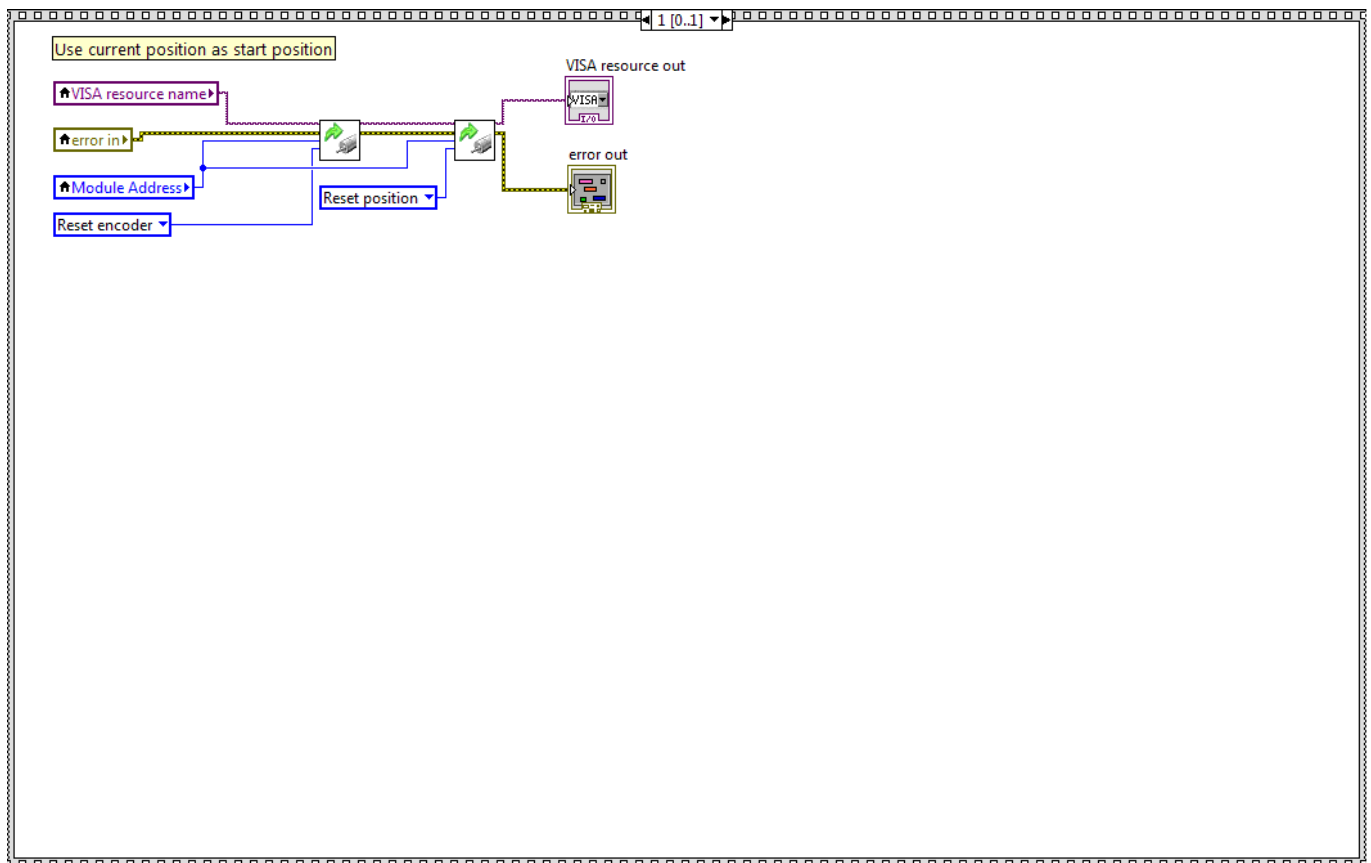


Don't rotate clockwise

False

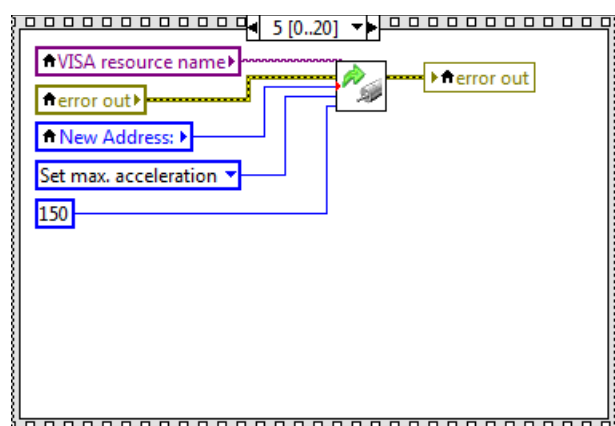
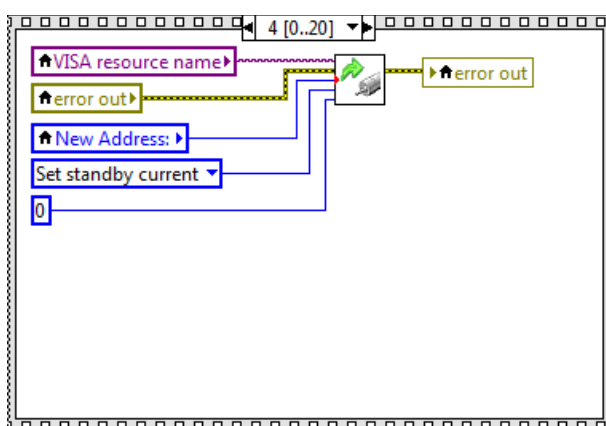
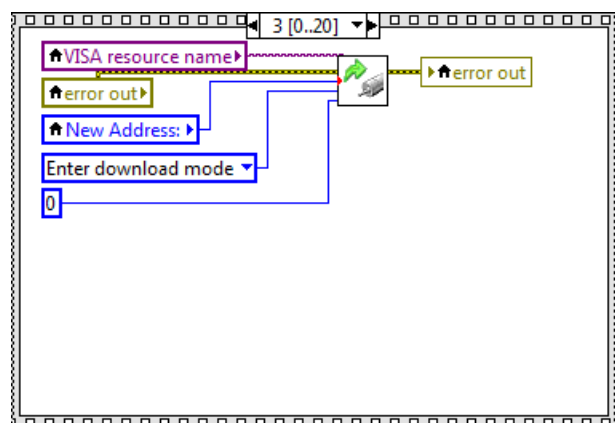
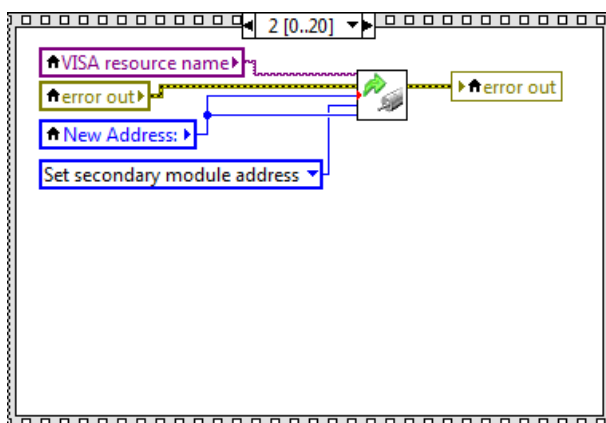
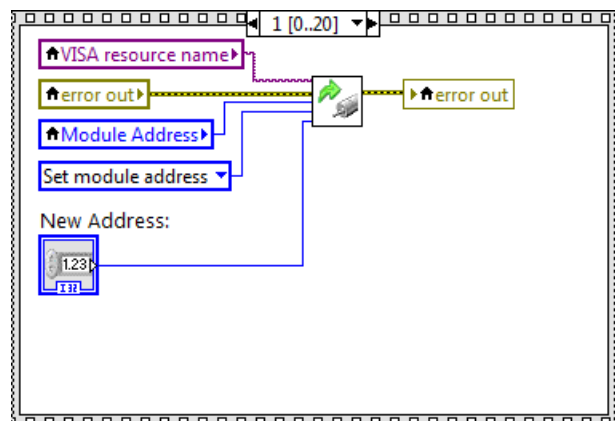
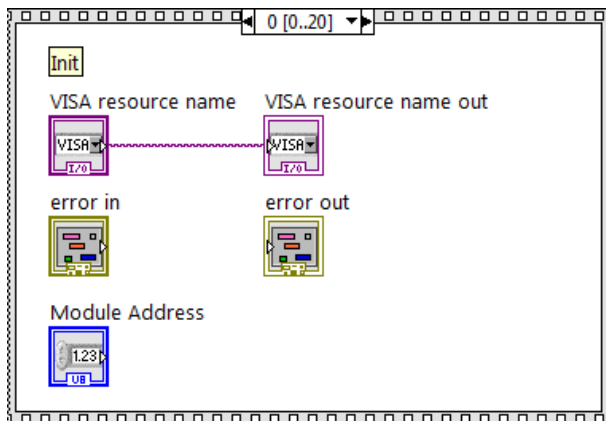
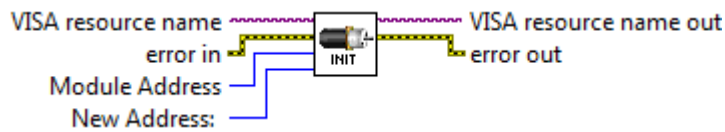
Don't rotate anti-clockwise

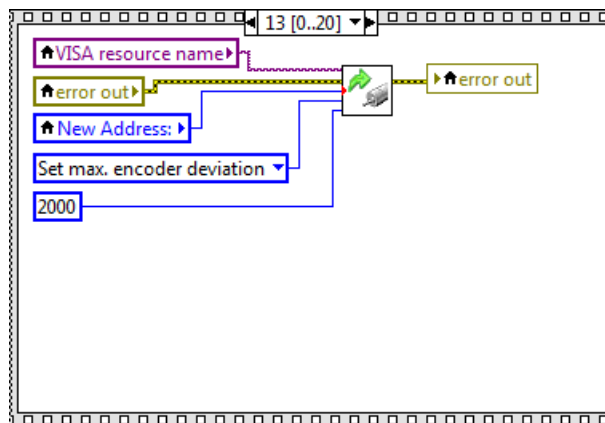
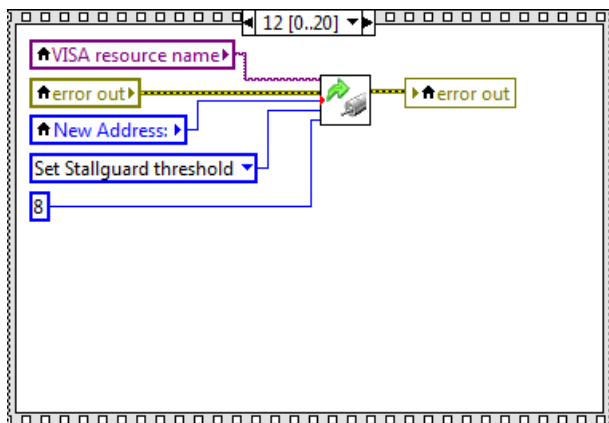
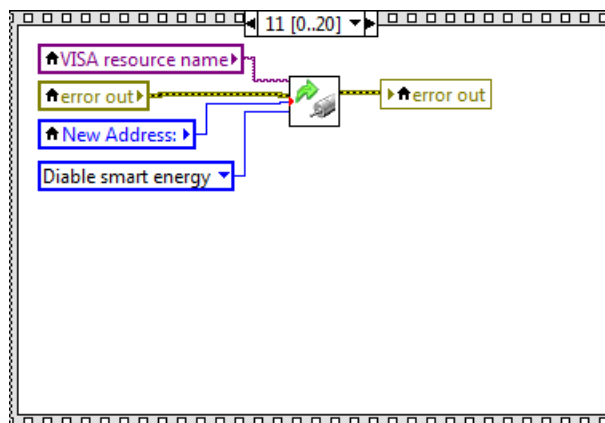
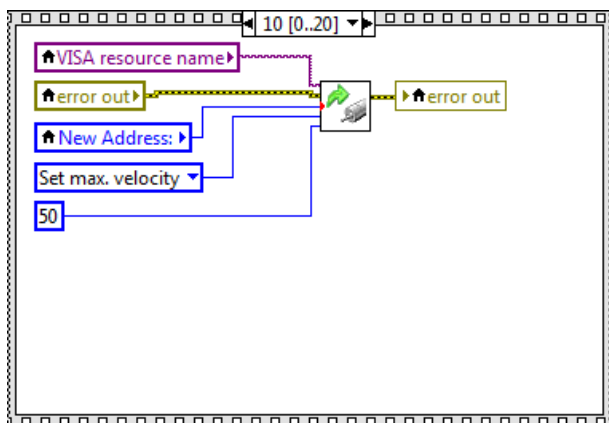
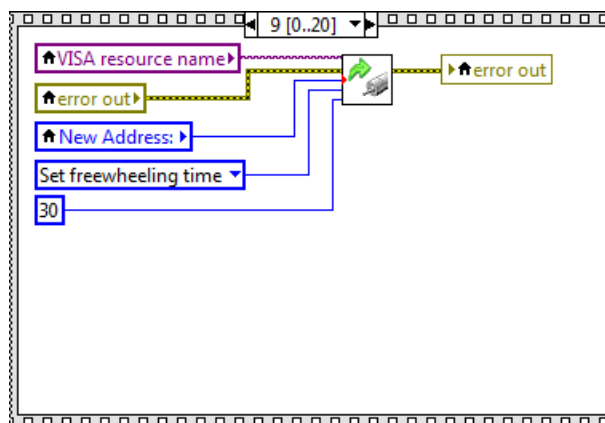
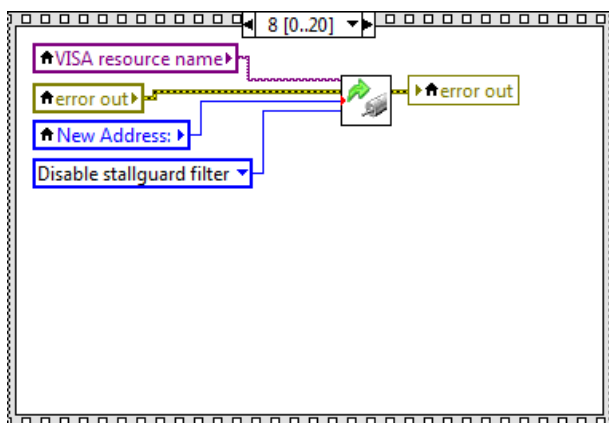
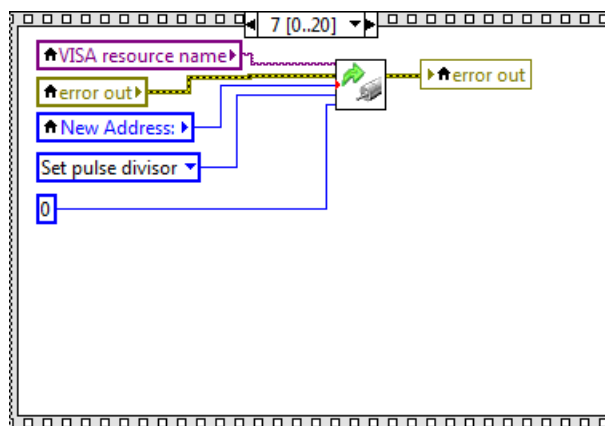
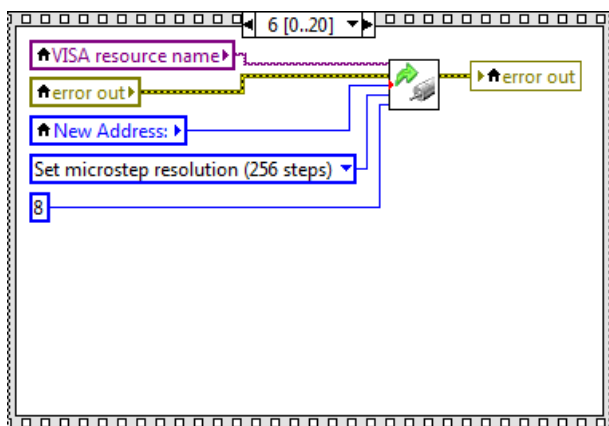
False

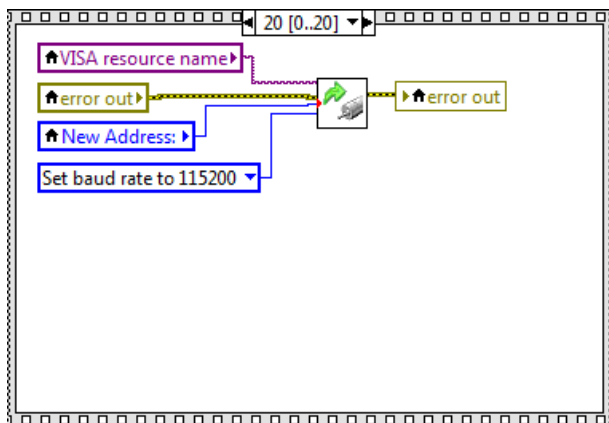
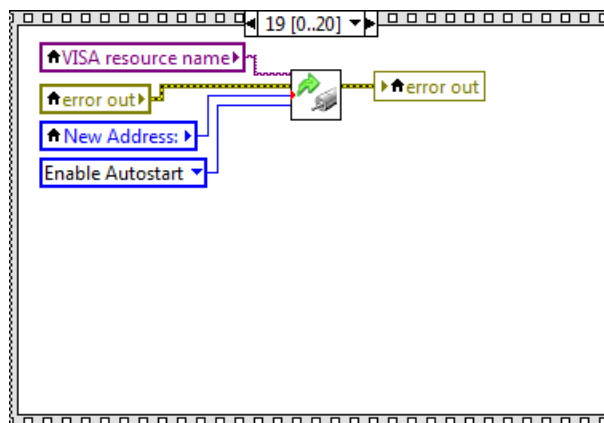
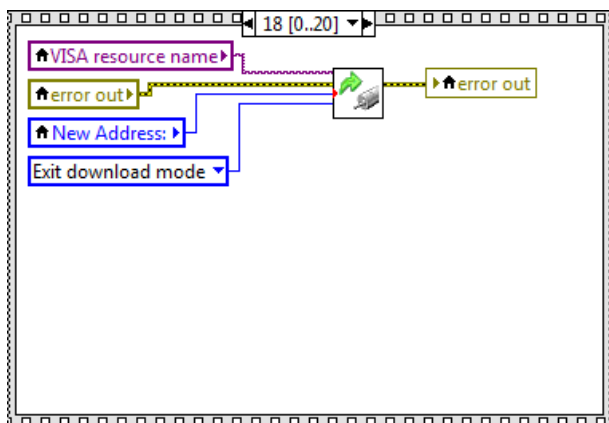
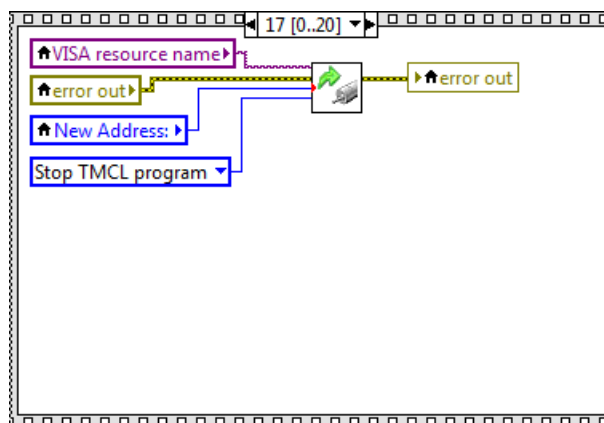
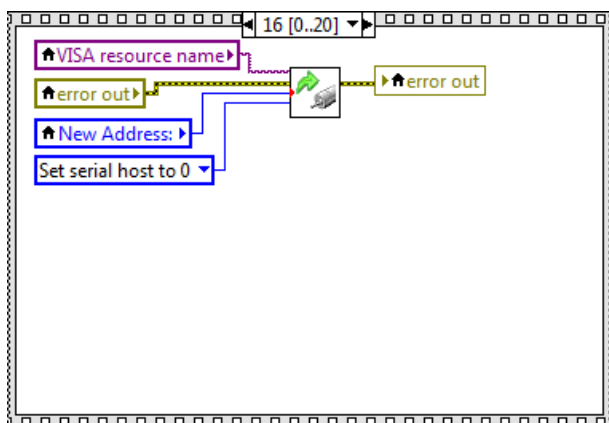
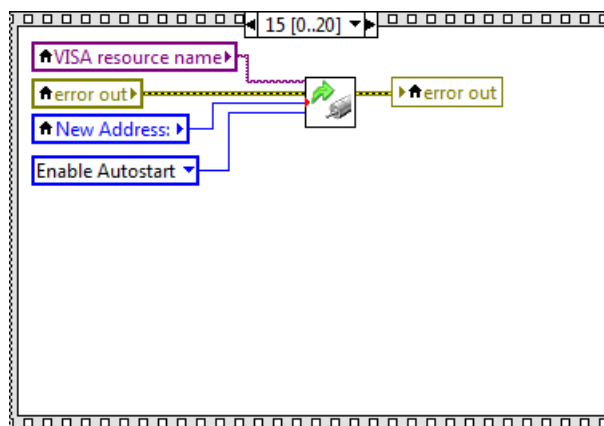
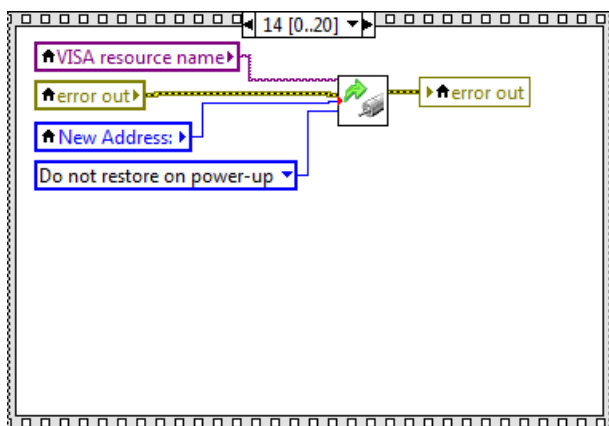


Configure_PANdrive_for_Switchy.vi

Sends init-program to PANdrive for compatibility with Rotary_switch software.

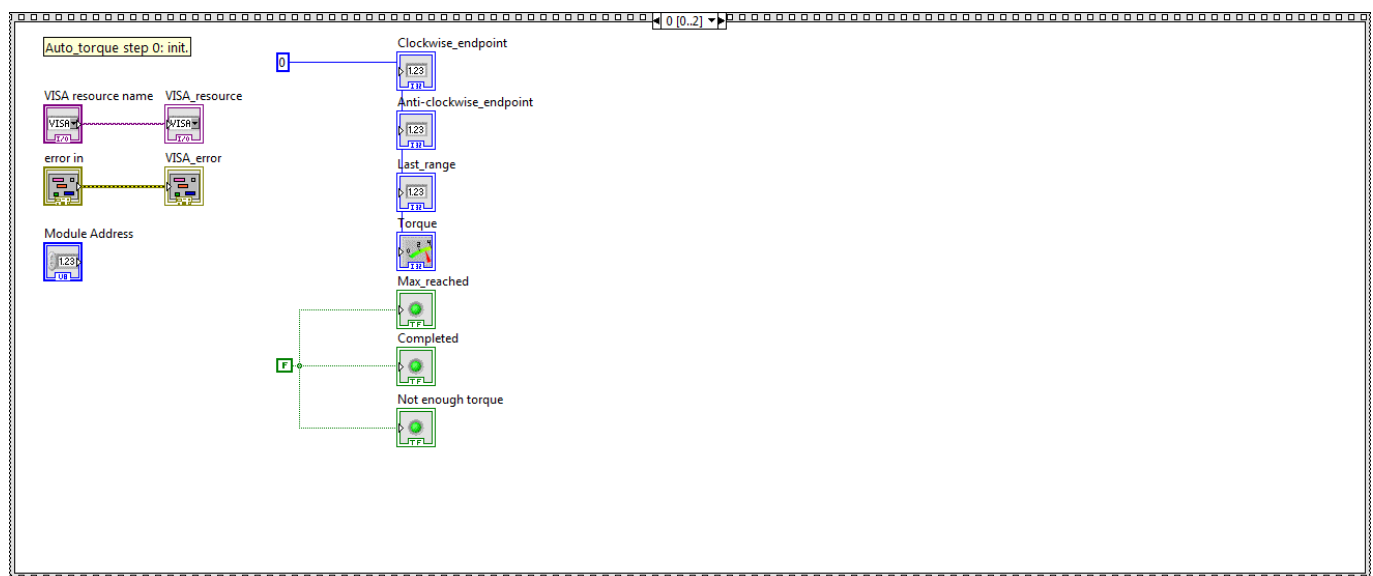
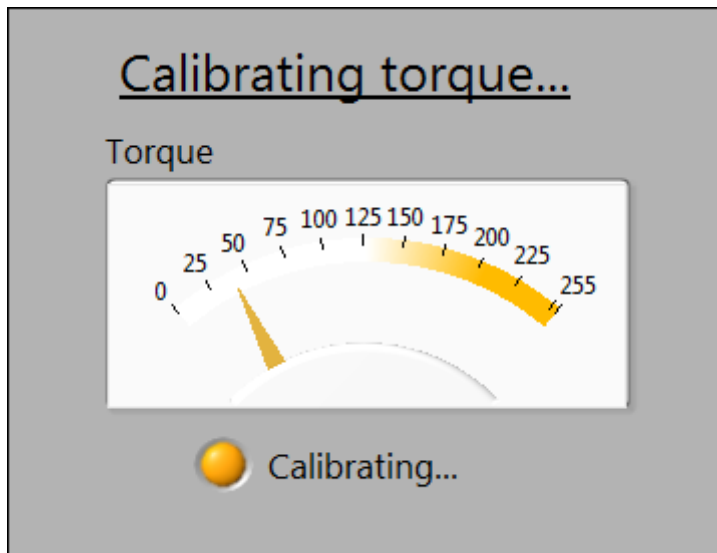
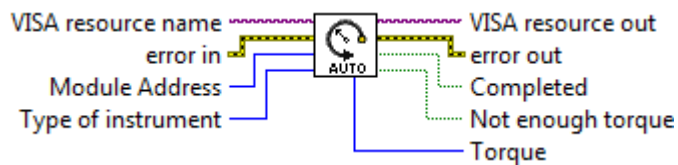


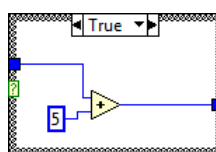
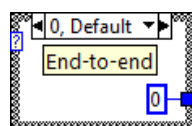
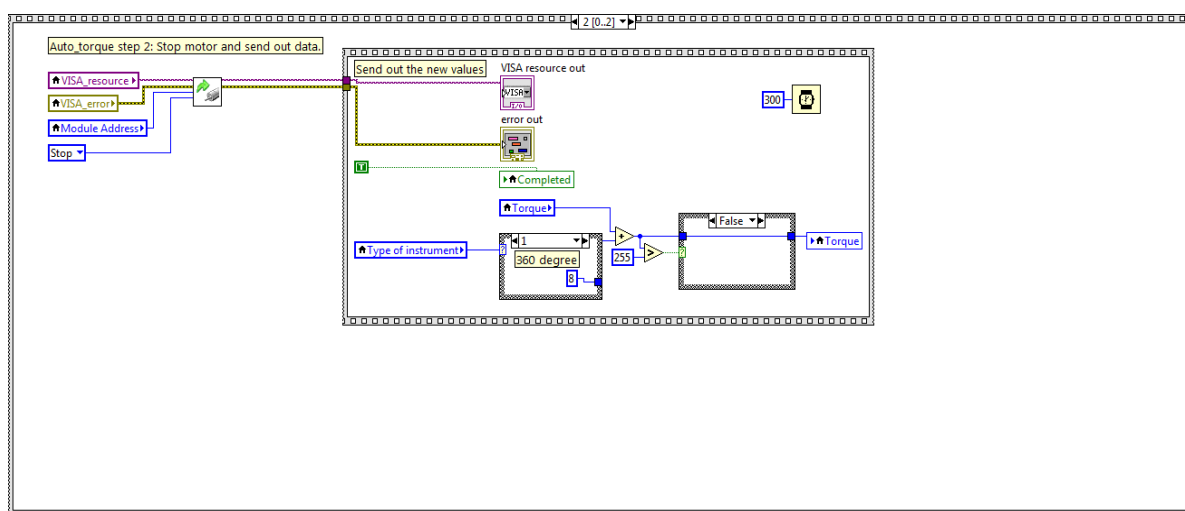
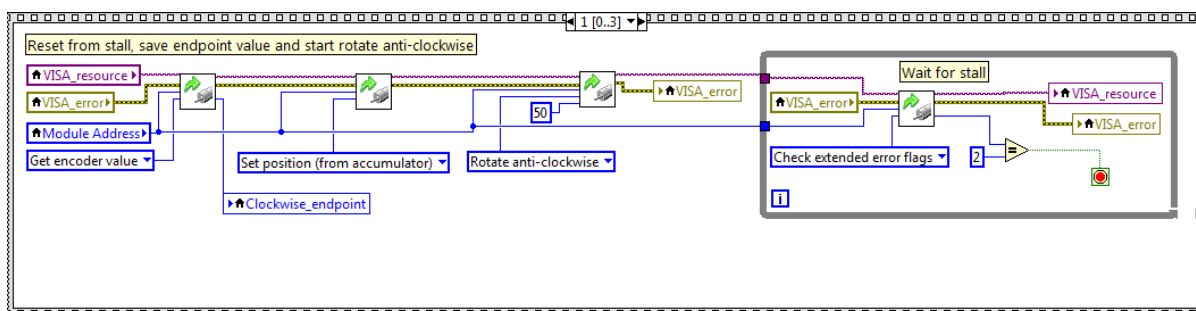
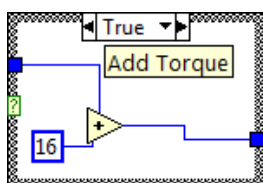
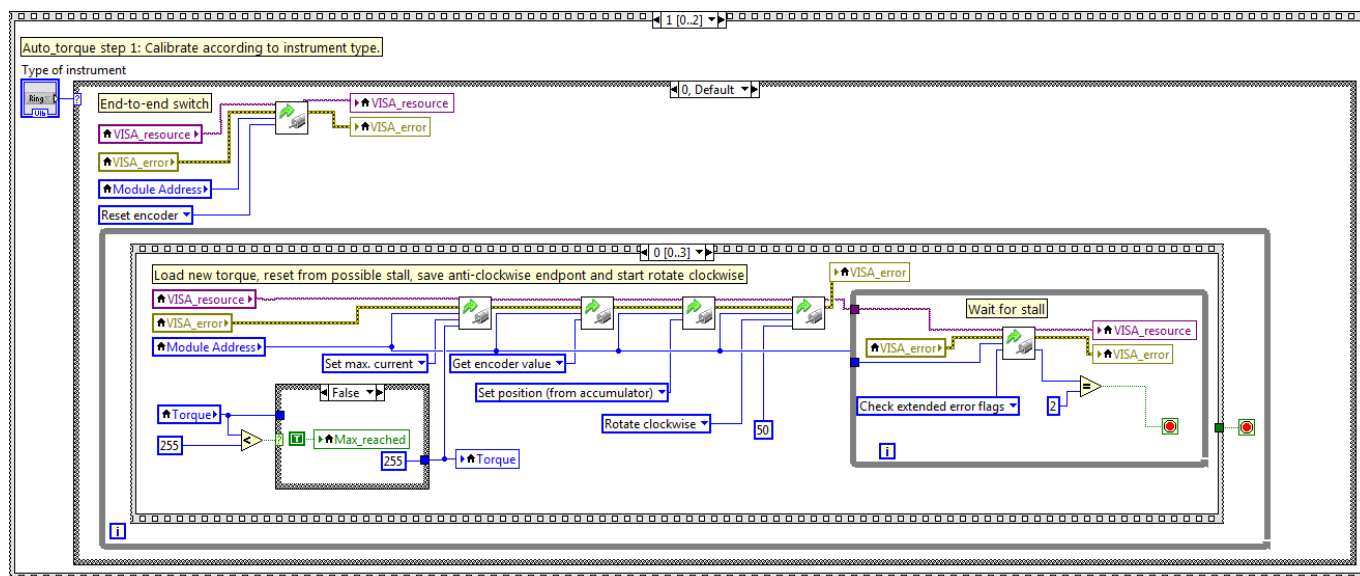


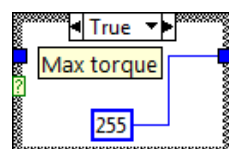
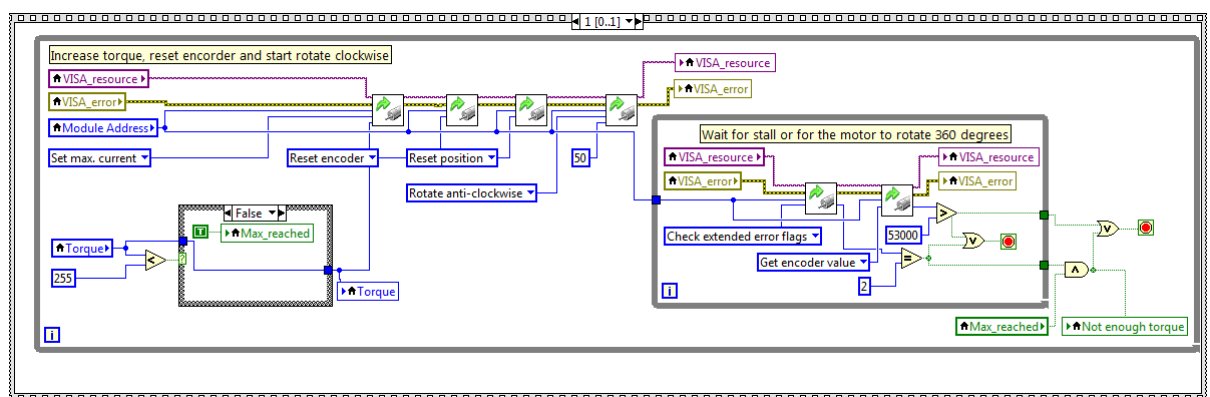
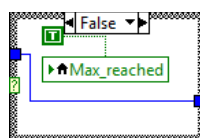
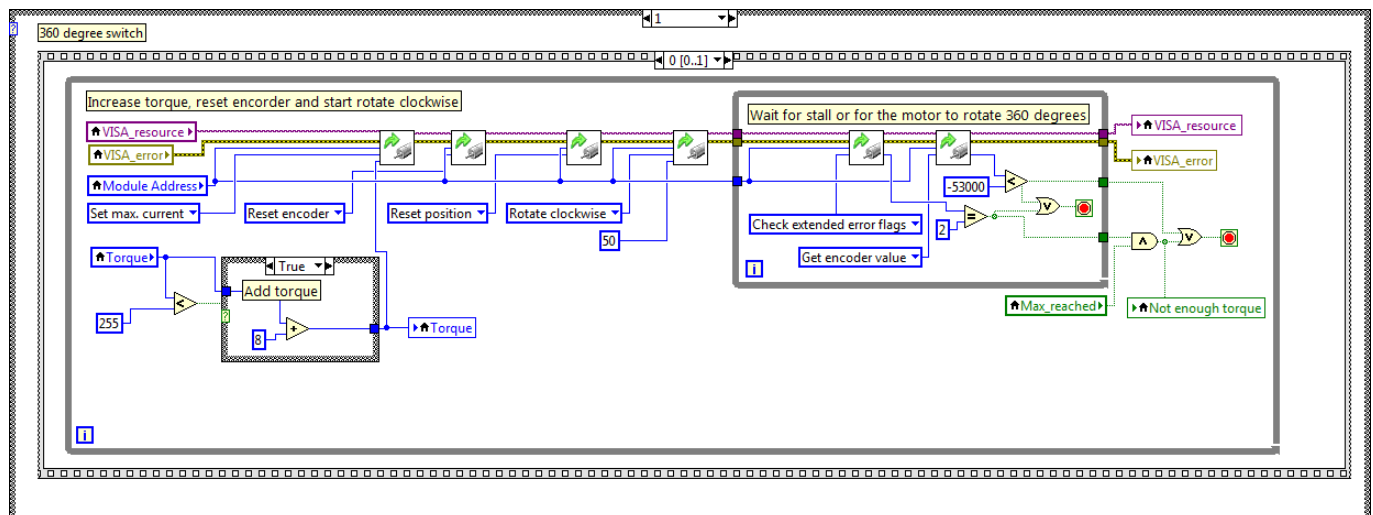
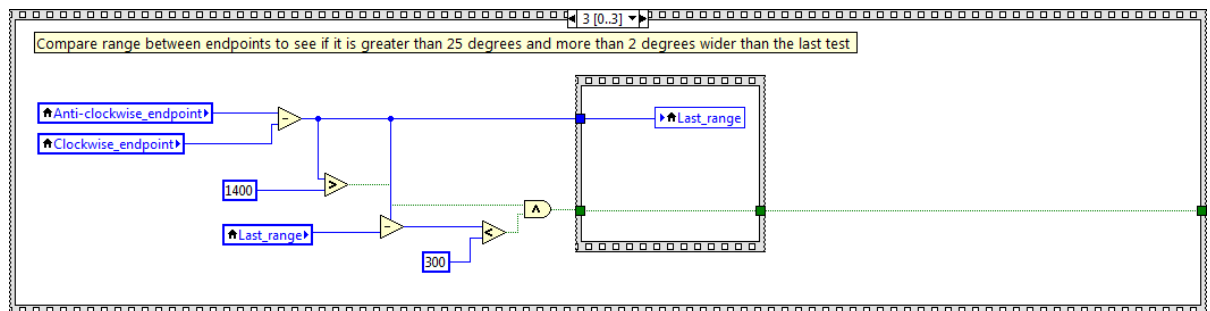
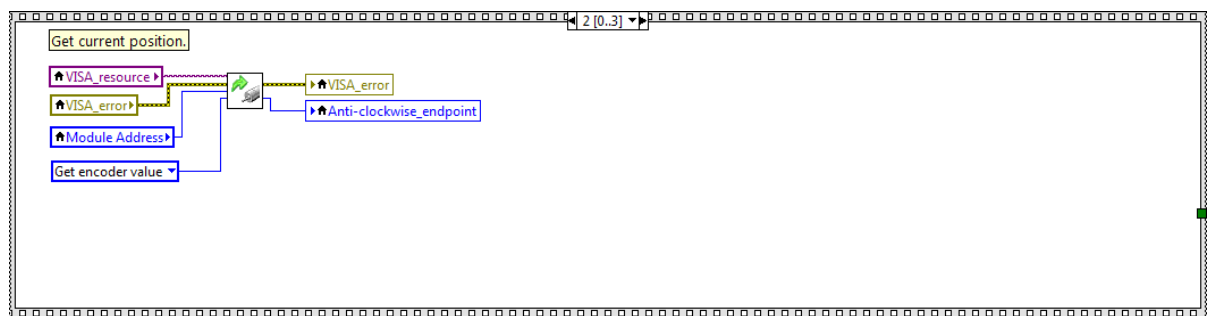


Auto_torque.vi

Automatic torque calibration. Finds the lowest torque that is needed to turn the switch and adds a small margin.

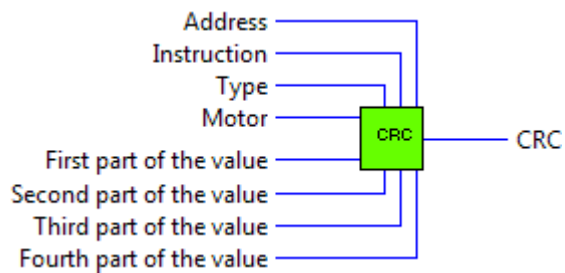






Calculate CRC.vi

Calculates checksum to send with an instruction to the PANdrive.



To get the CRC value we have to add all 8 bytes.

Address



Instruction



Type



Motor



First part of the value



Second part of the value



Third part of the value



Fourth part of the value

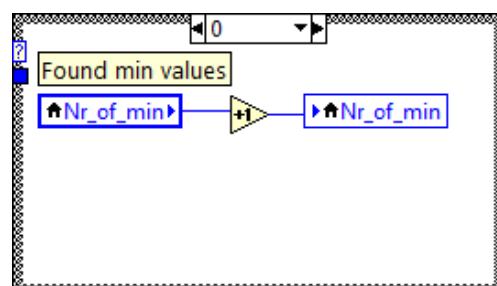
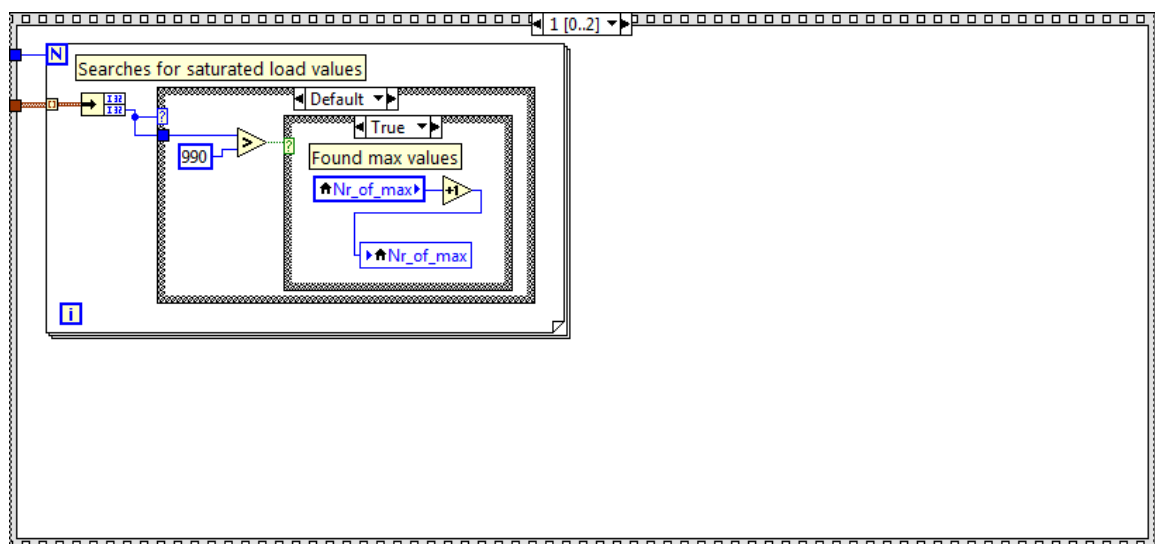
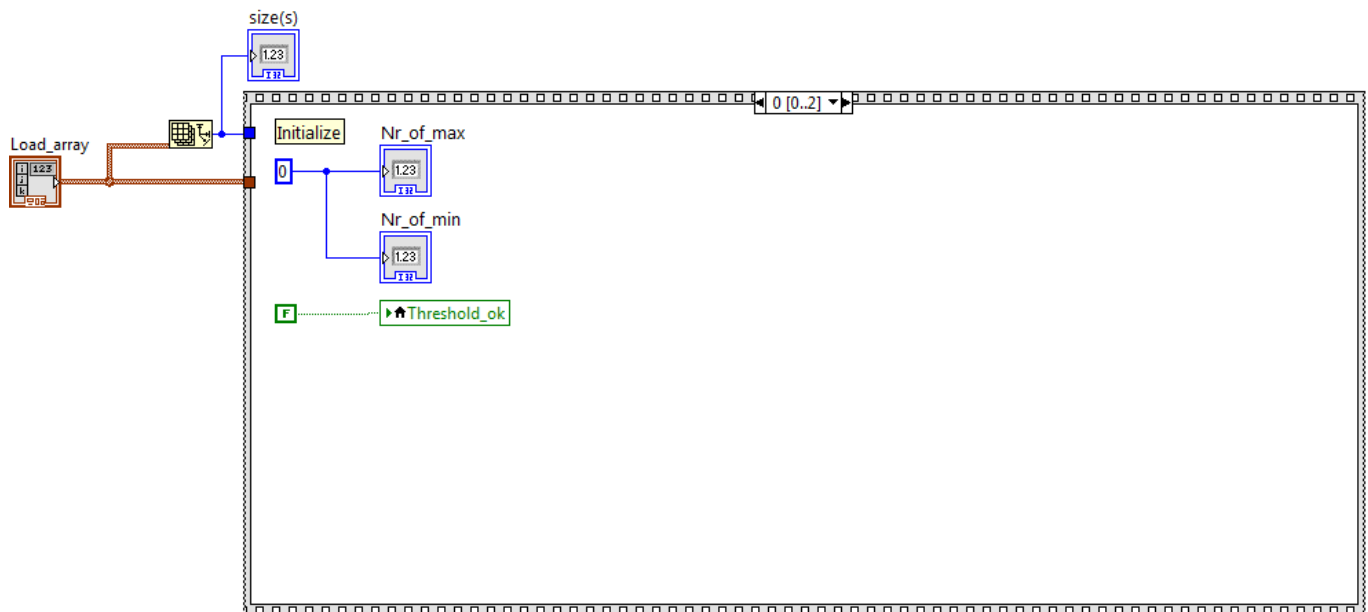
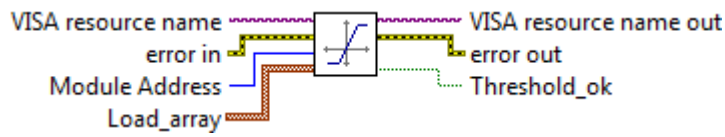


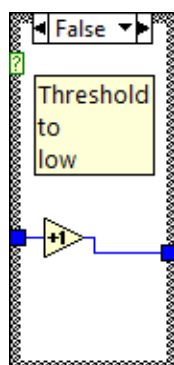
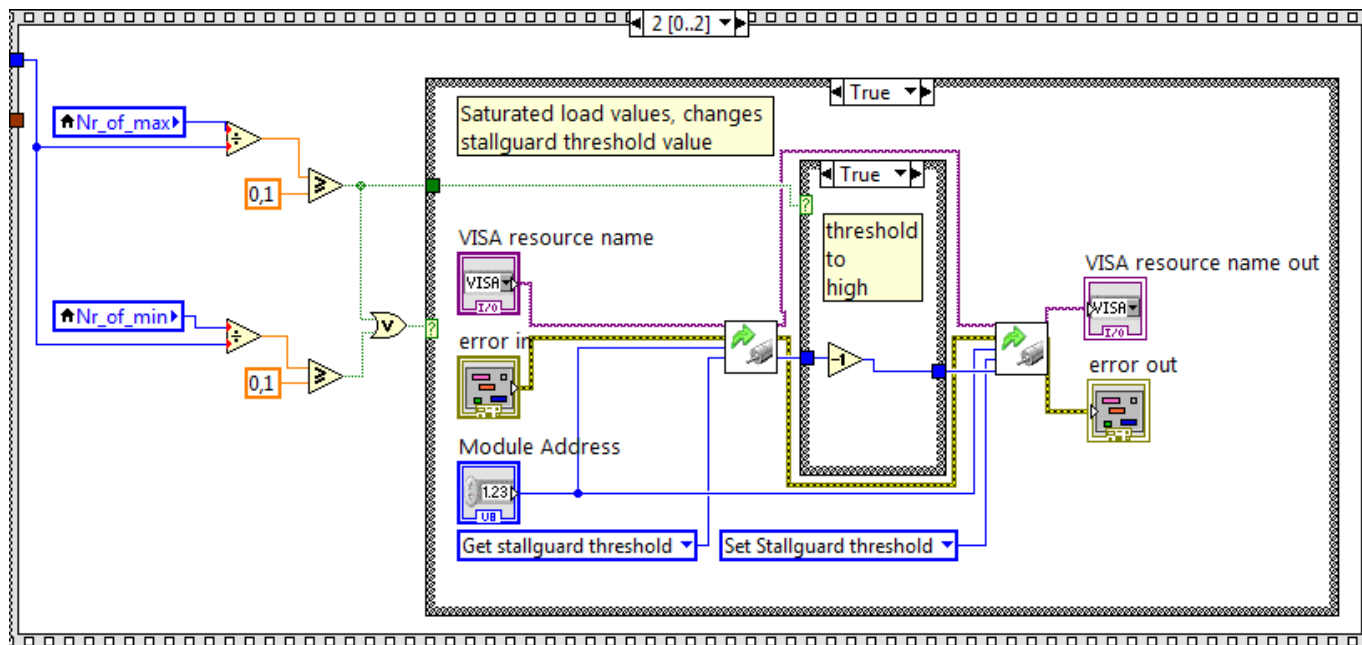
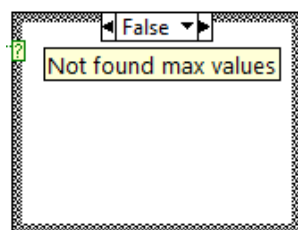
CRC



Check_threshold.vi

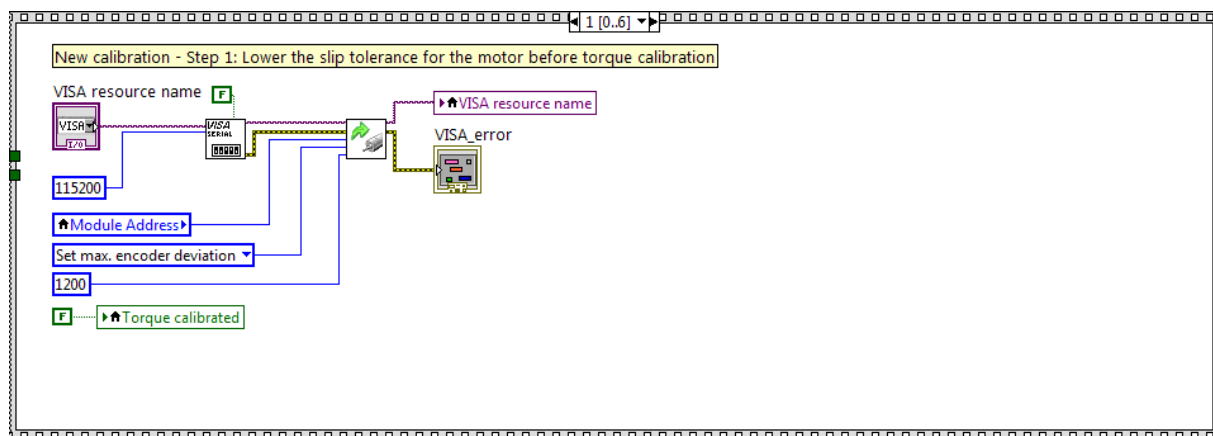
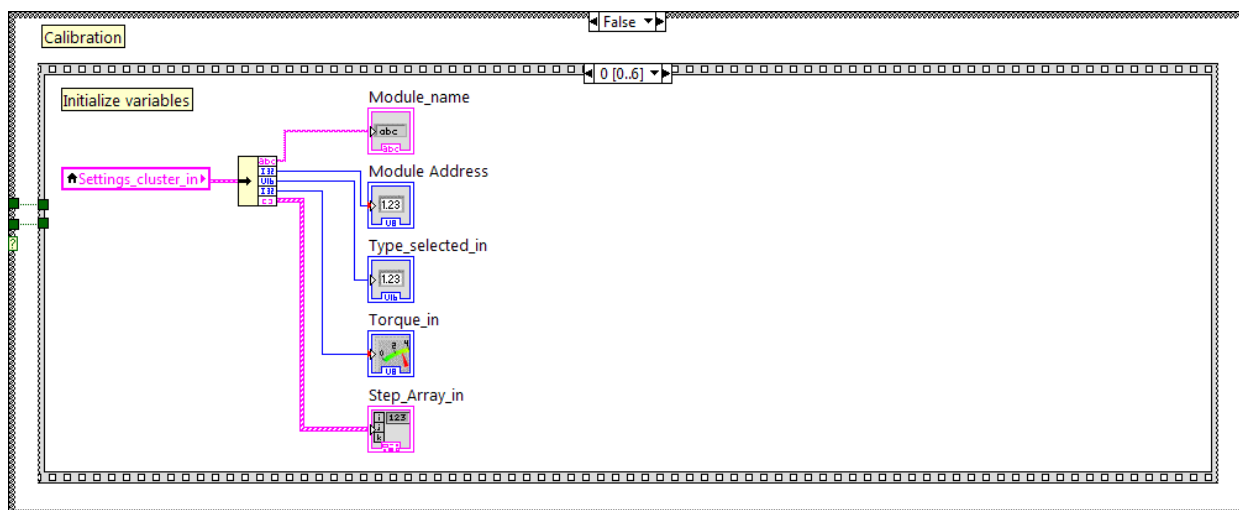
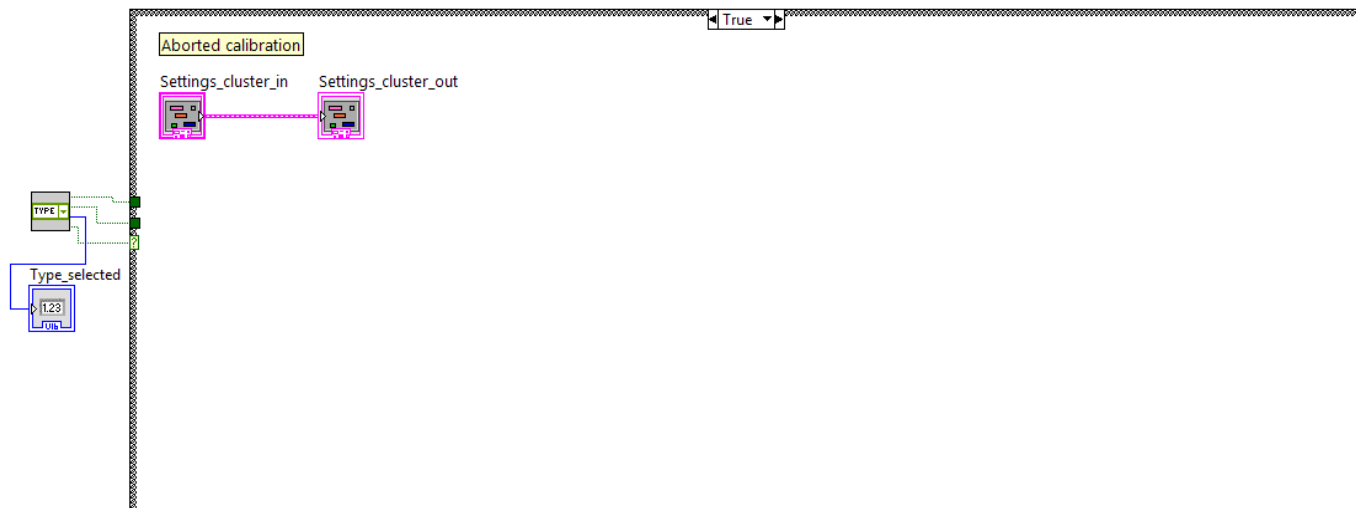
Checks if the current stallguard threshold gives usable values. Otherwise increases or decreases it.

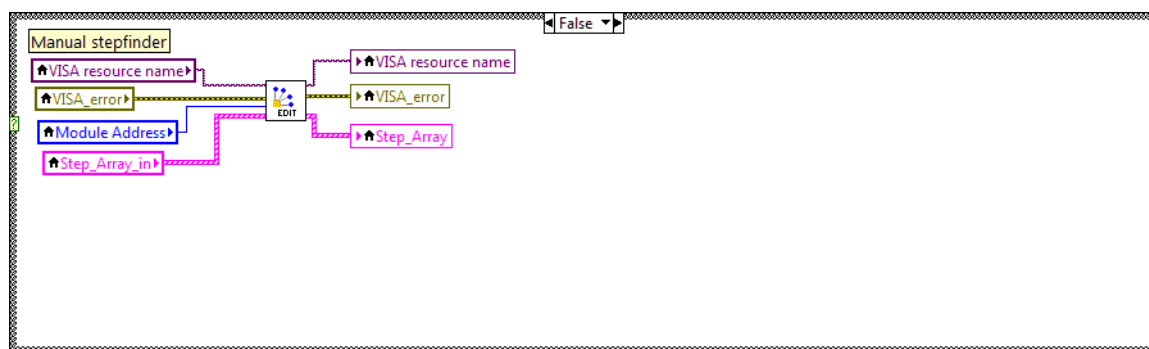
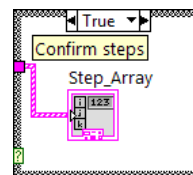
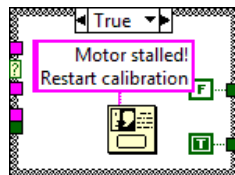
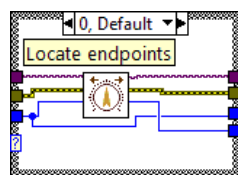
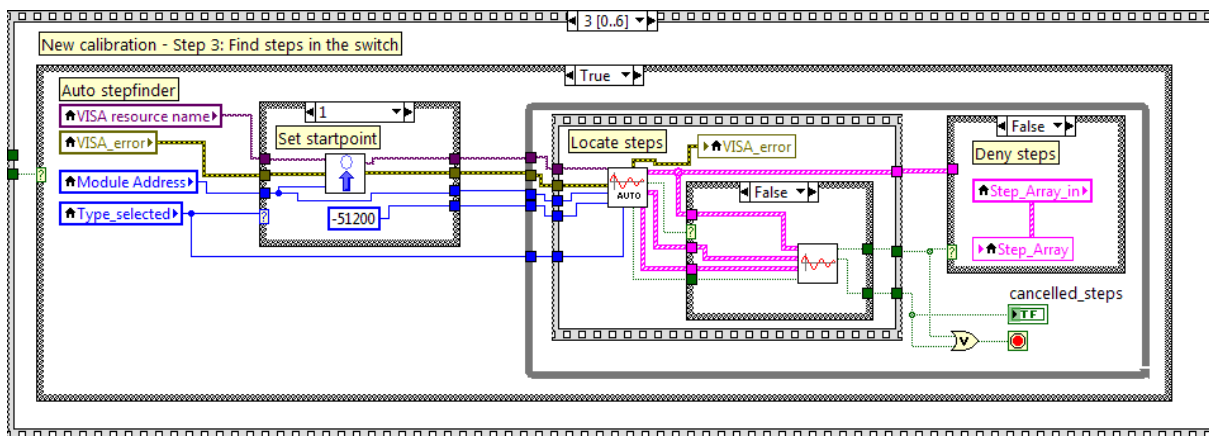
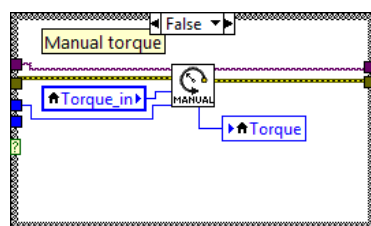
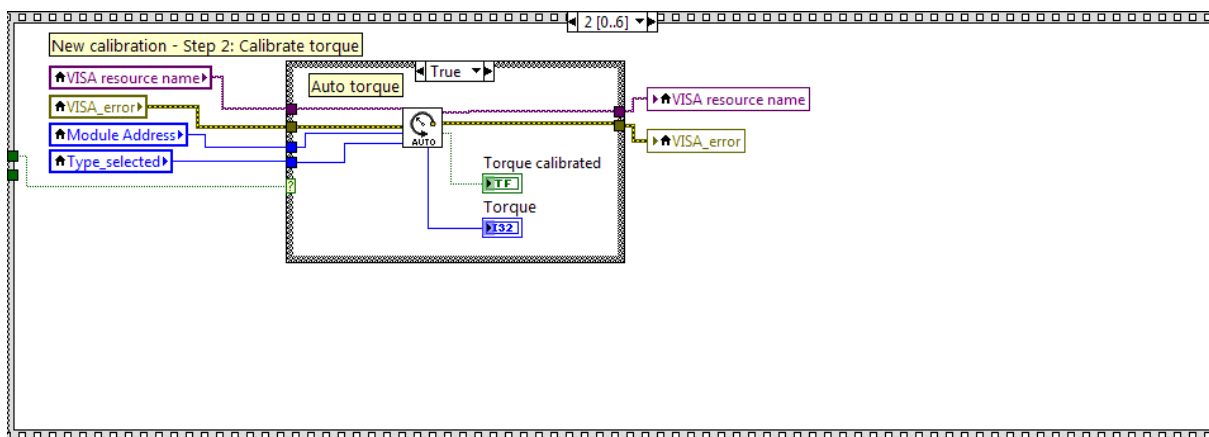


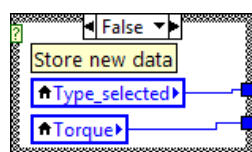
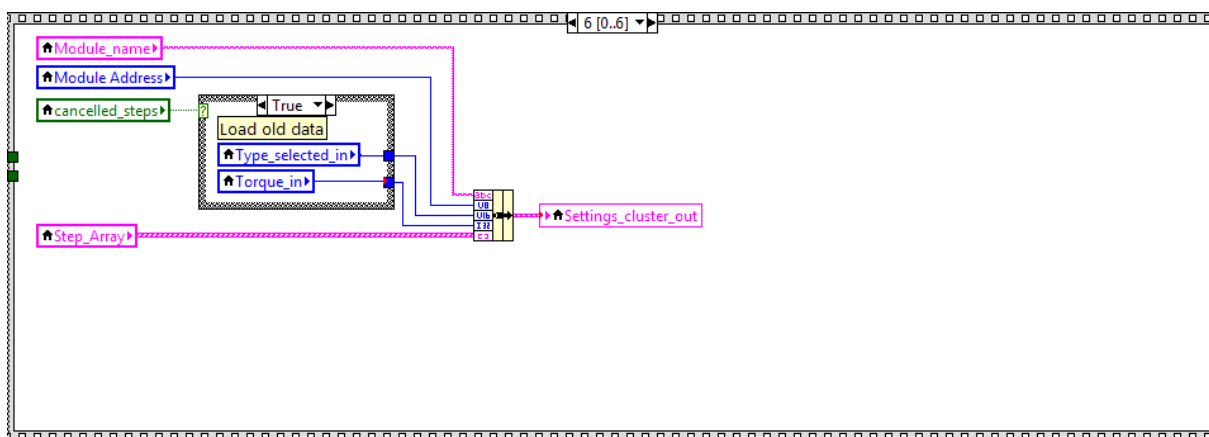
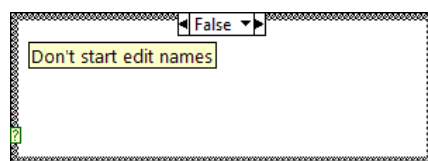
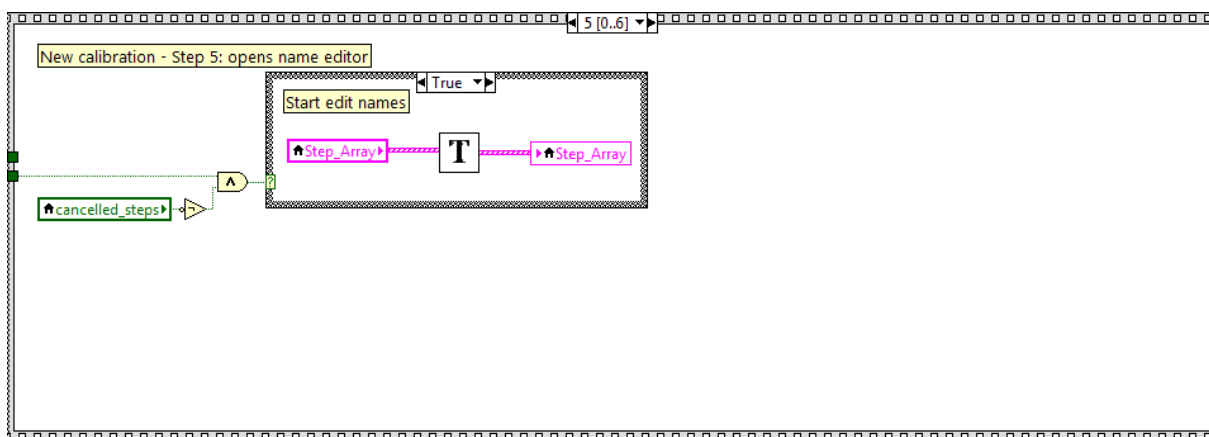
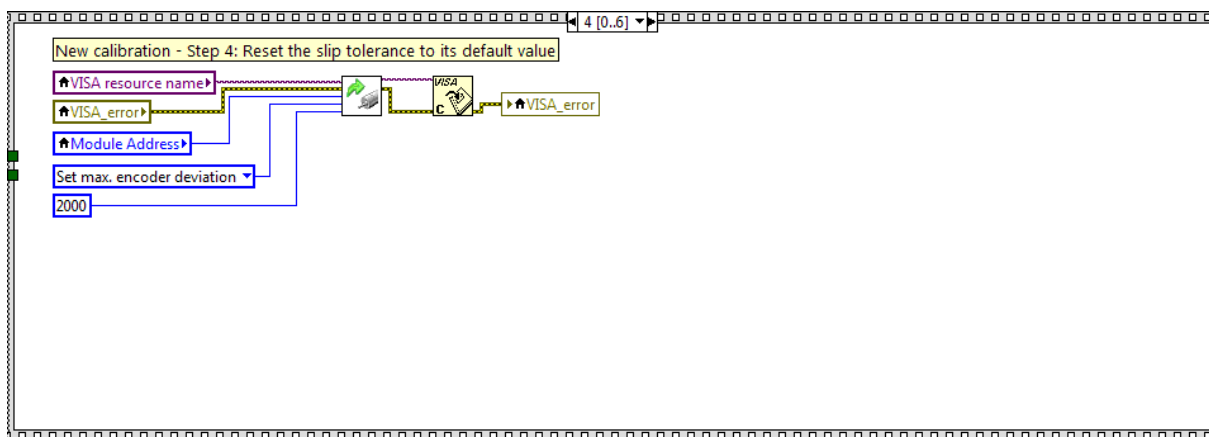


Configure_new.vi

Starts new rotary switch calibration.

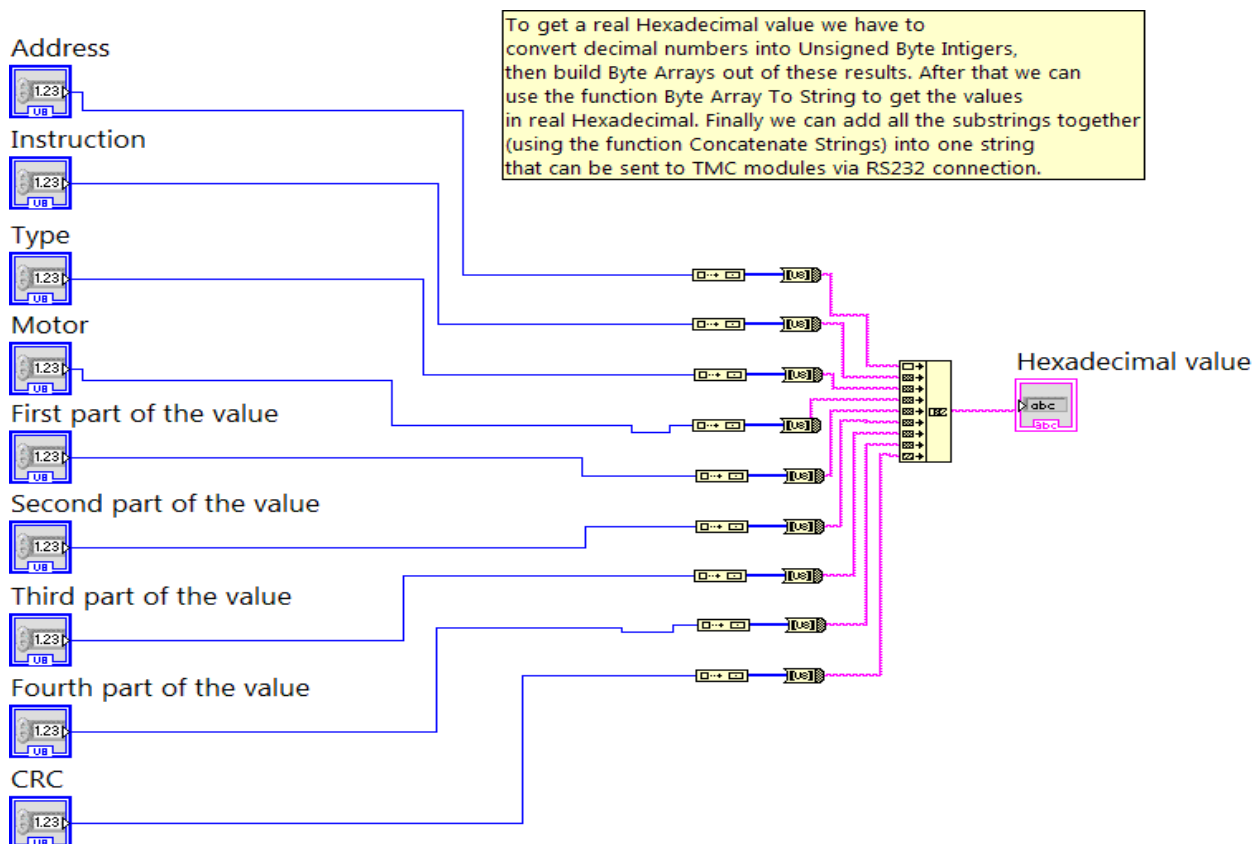
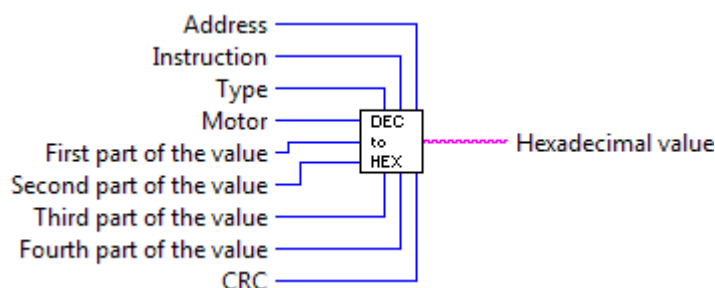






Decimal to hex.vi

Combines decimal numbers to a single hexadecimal number. This value will be sent as an instruction to the PANDrive.



Edit_names.vi

Let's the users set names for steps found using automatic stepfinder.

Step_Array_in  Step_Array_out

Name Editor

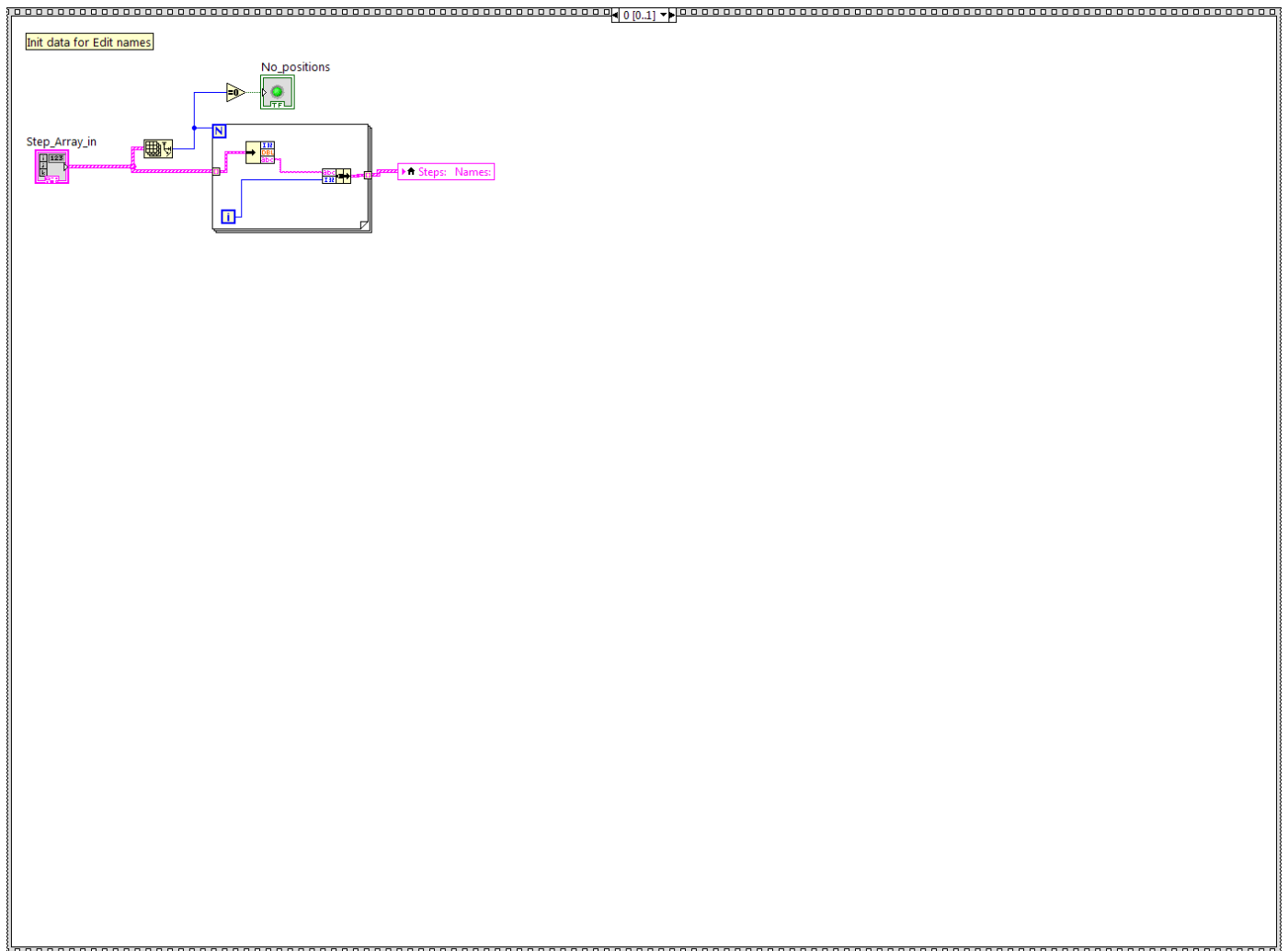
Steps: Names:

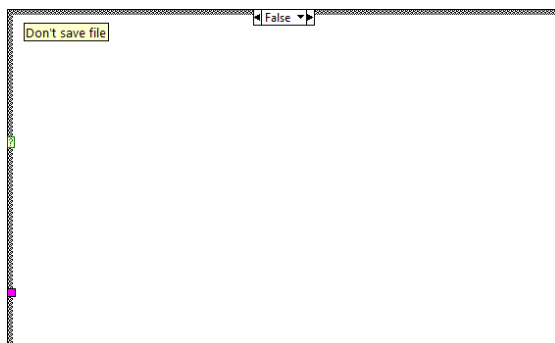
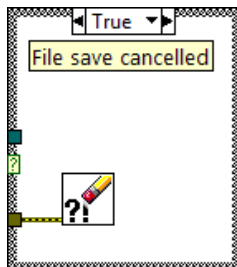
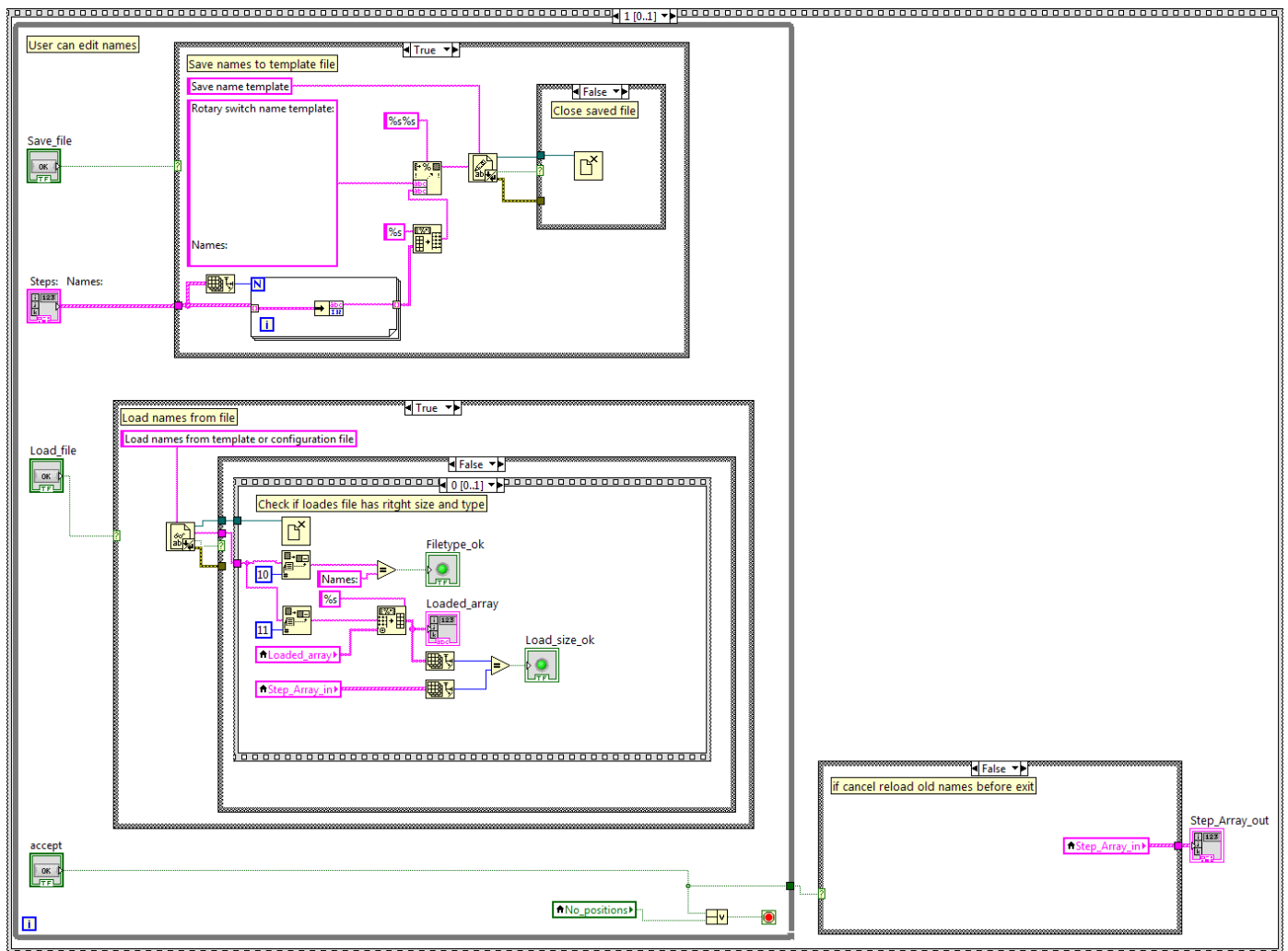
0	
0	
0	
0	
0	
0	
0	
0	
0	
0	
0	
0	
0	
0	

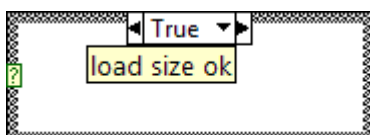
SAVE/LOAD NAME TEMPLATE

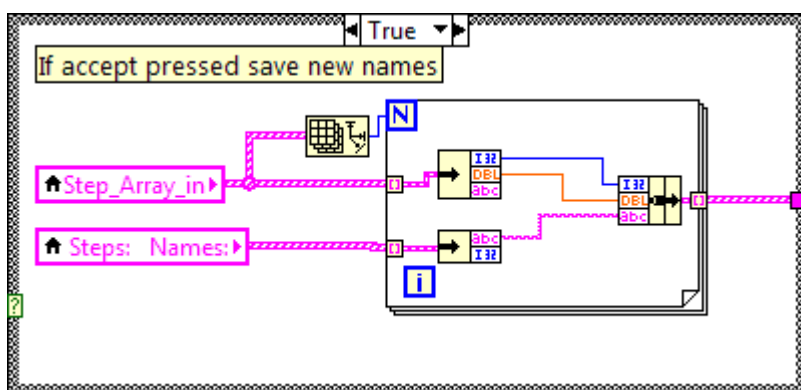
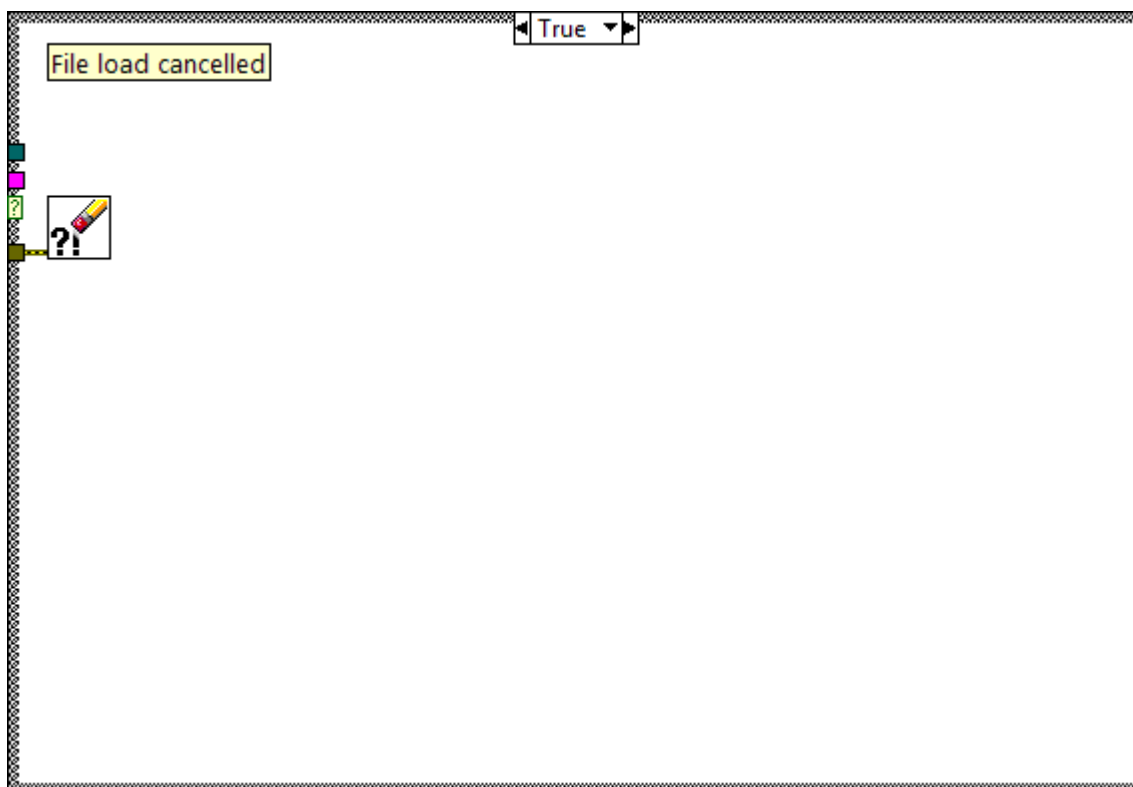
SAVELOAD

 ACCEPT









Edit_steps_found.vi

Lets user edit step configuration after initial calibration. Step positions and names can be changed, and steps can be ramove/added.

VISA resource name  VISA_error
Settings_cluster_in  Settings_cluster_out

Step Editor

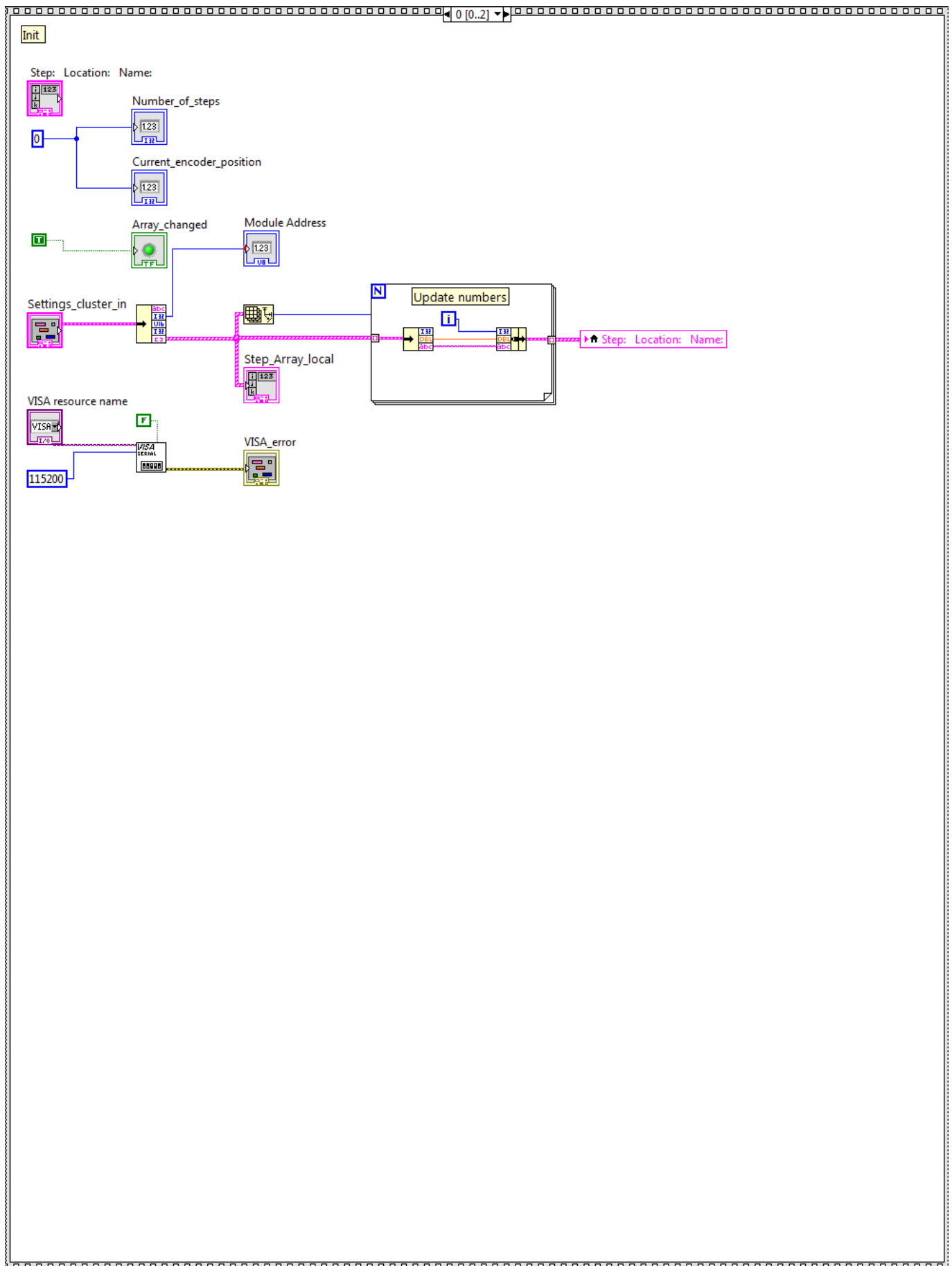
Step:	Location:	Name:
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	
0	0,0	

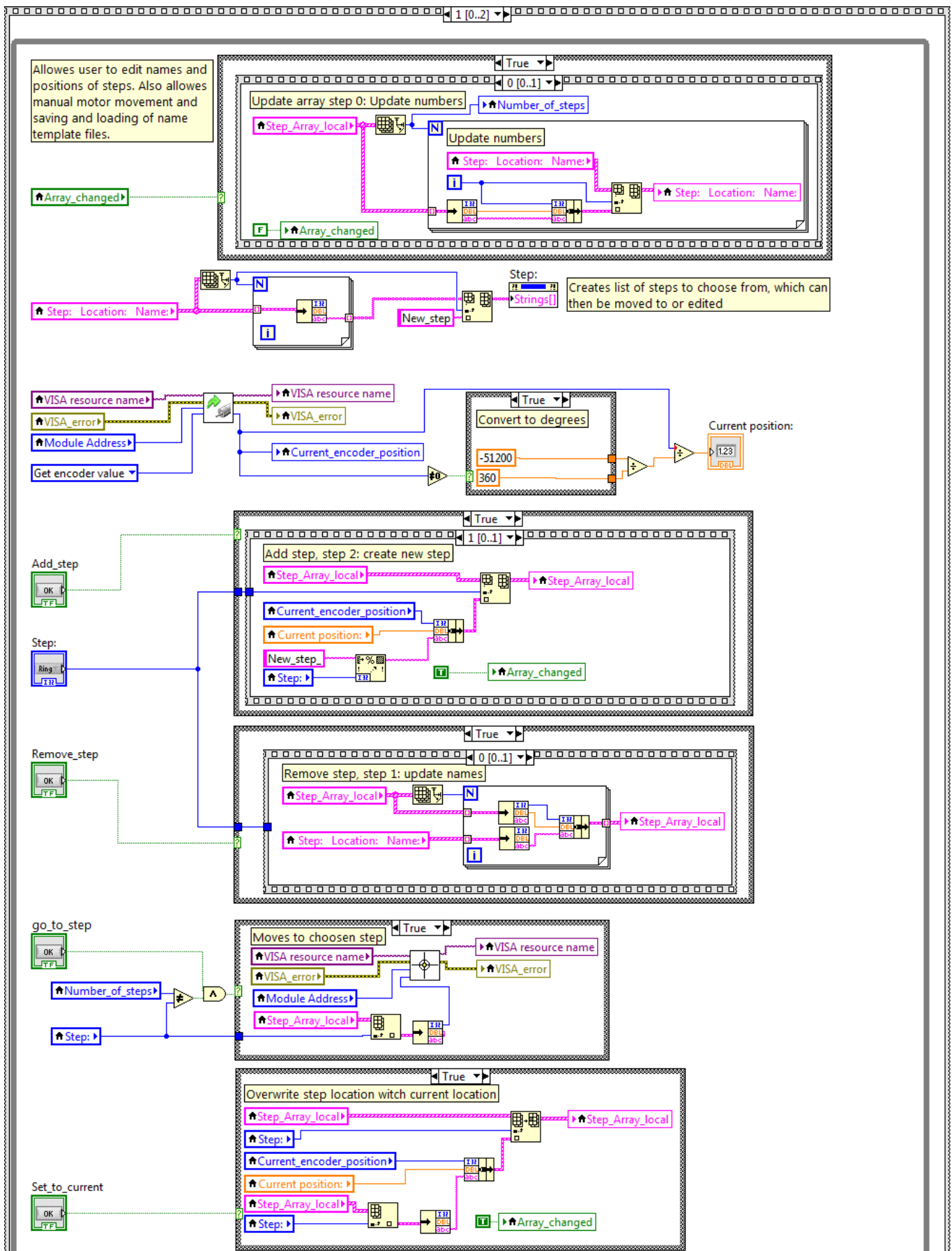
Current position:

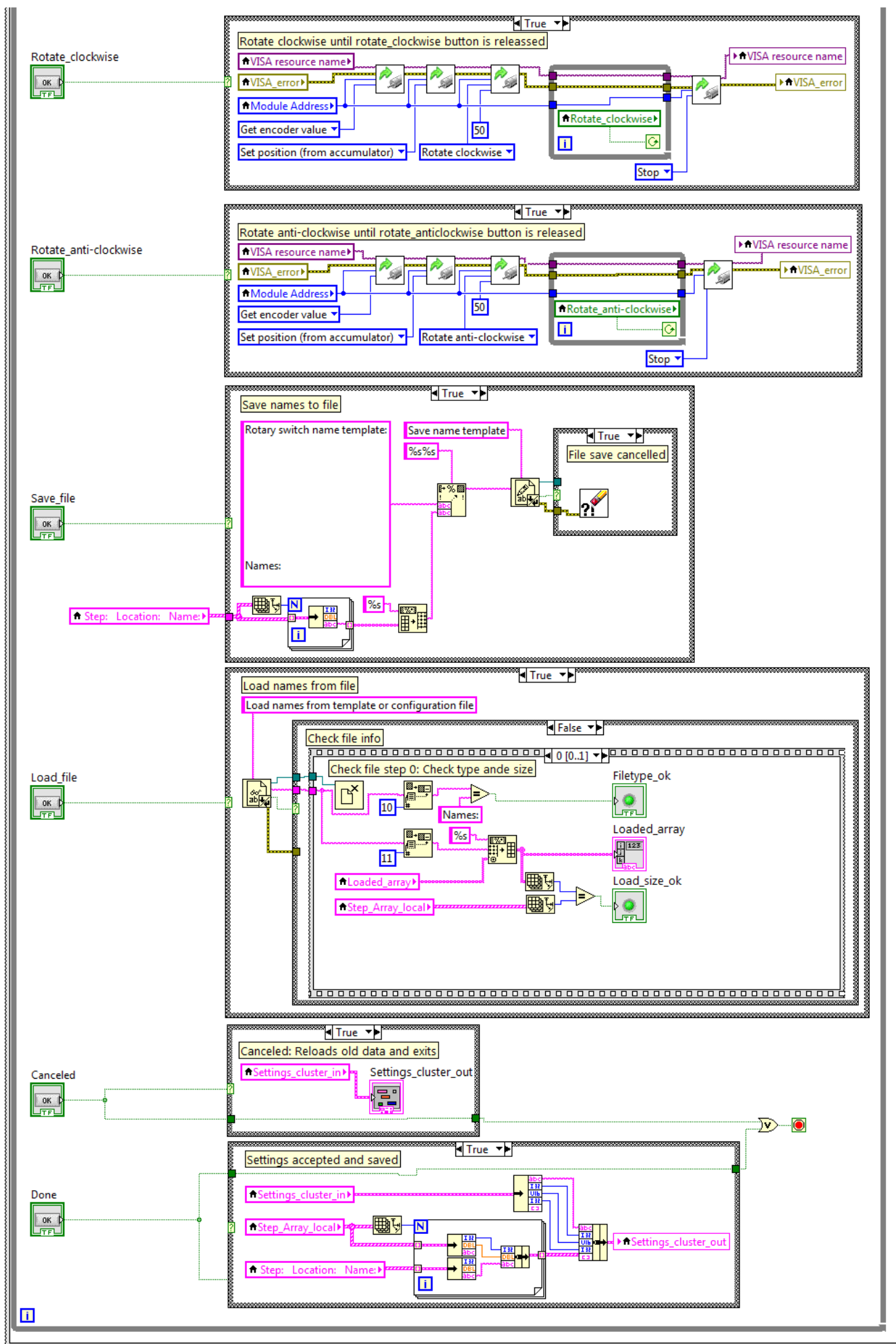
EDIT STEP POSITIONS

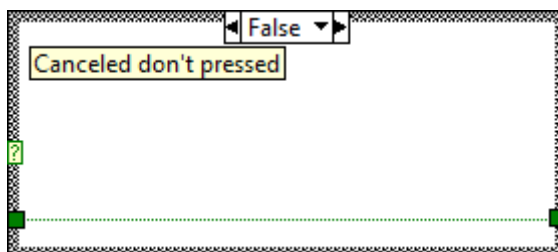
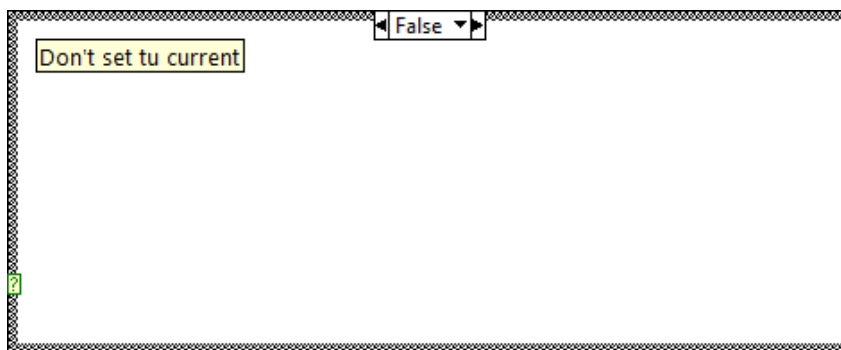
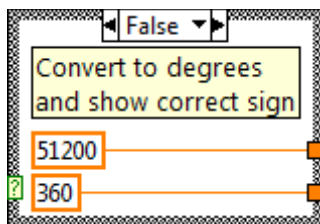
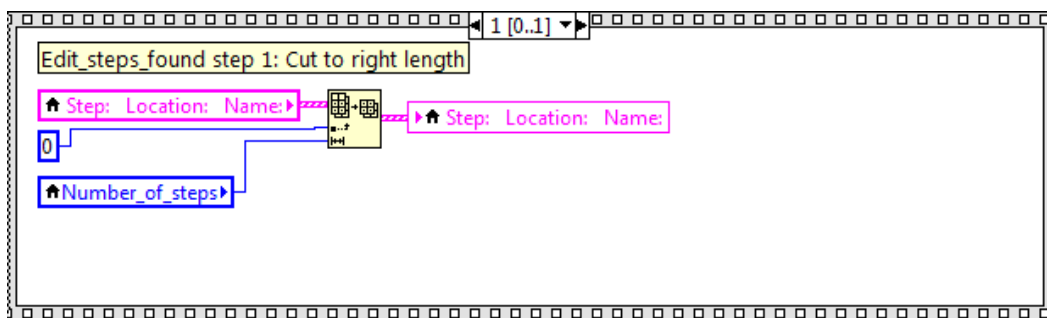
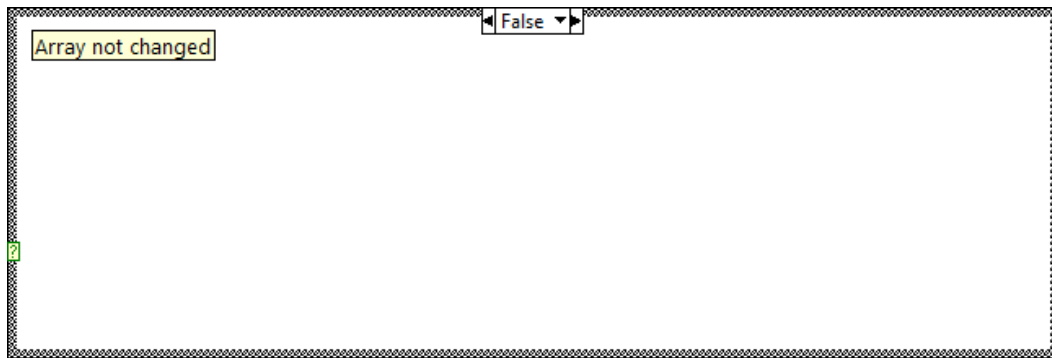
Step:

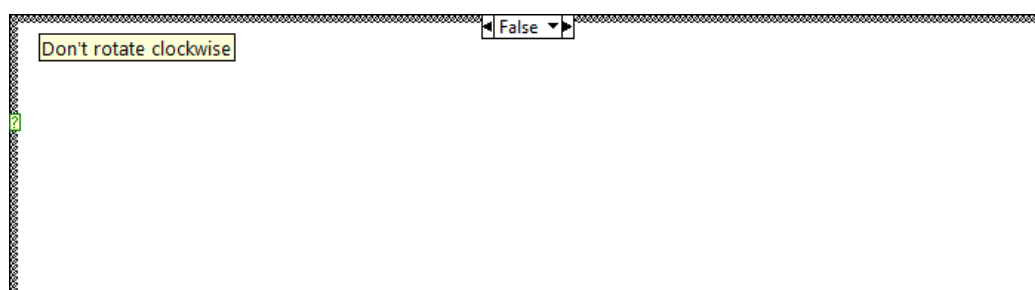
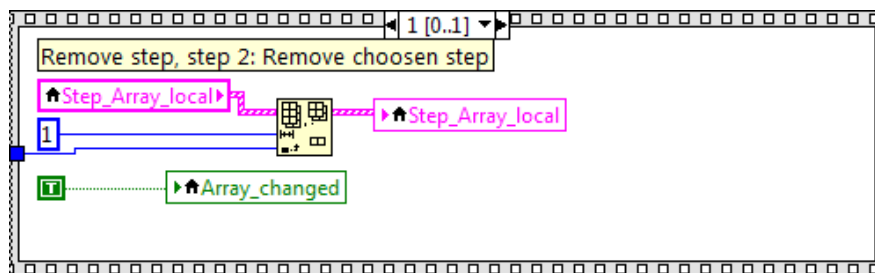
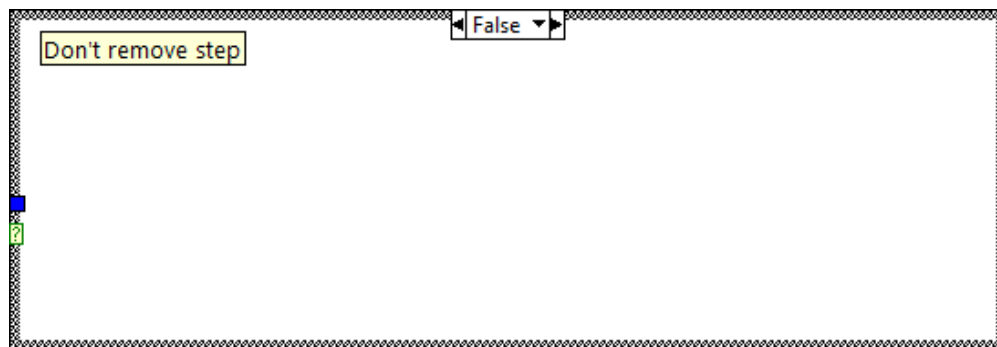
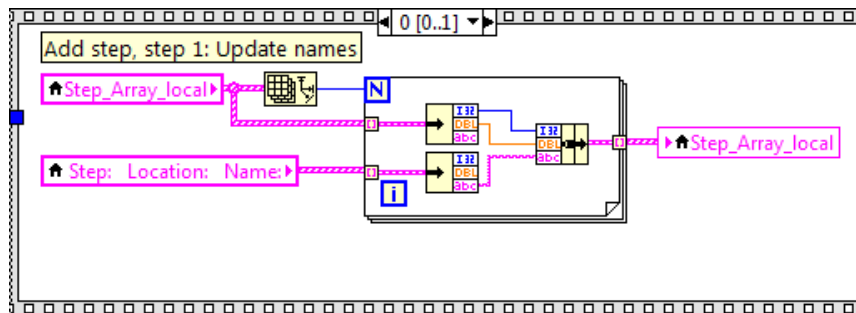
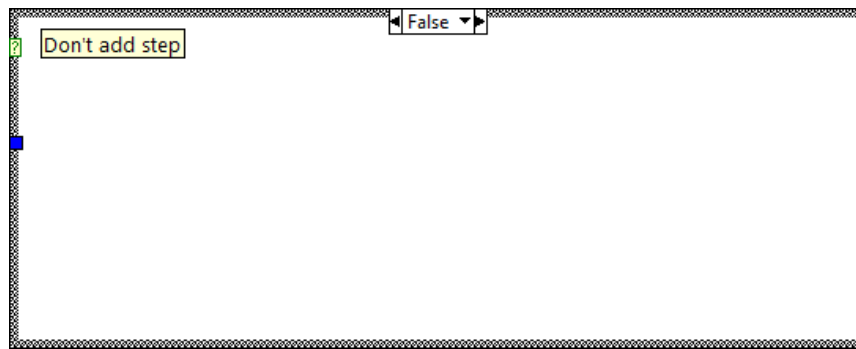
SAVE/LOAD NAME TEMPLATE

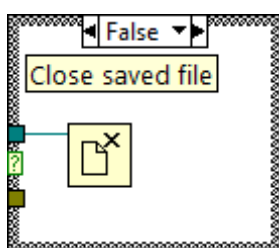
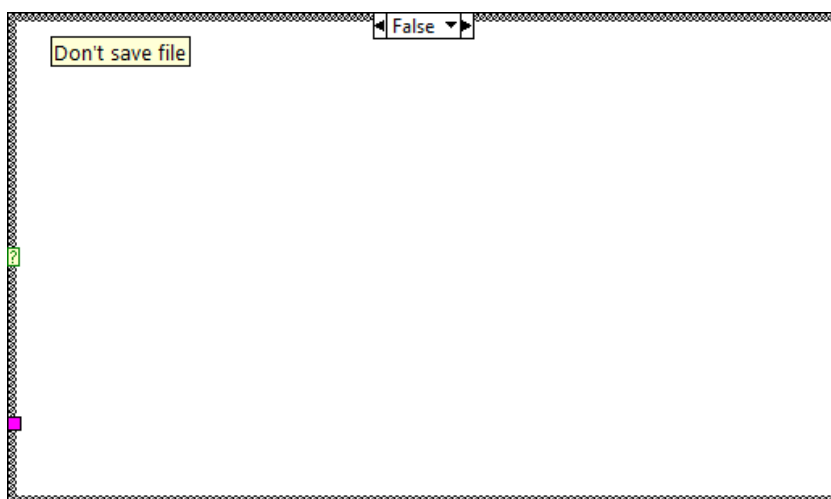
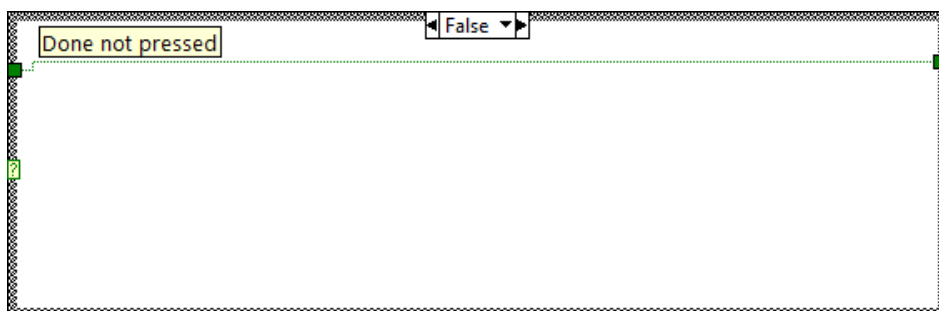
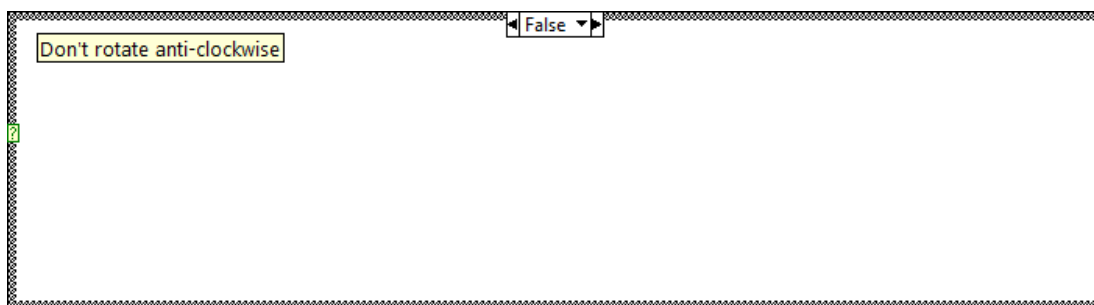


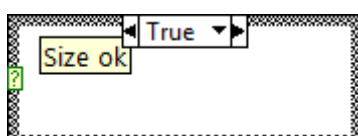
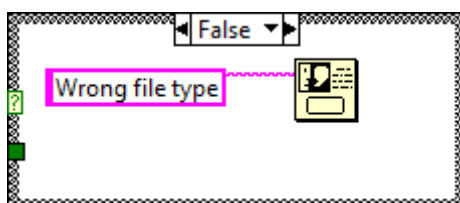
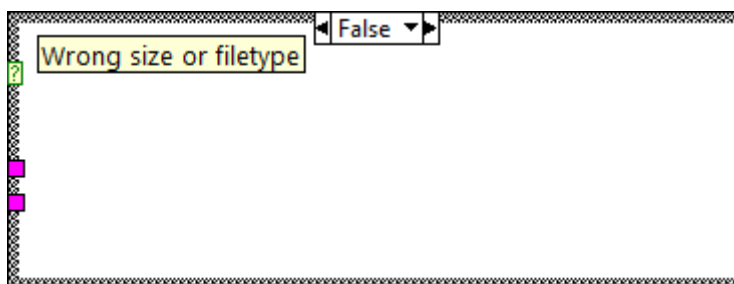
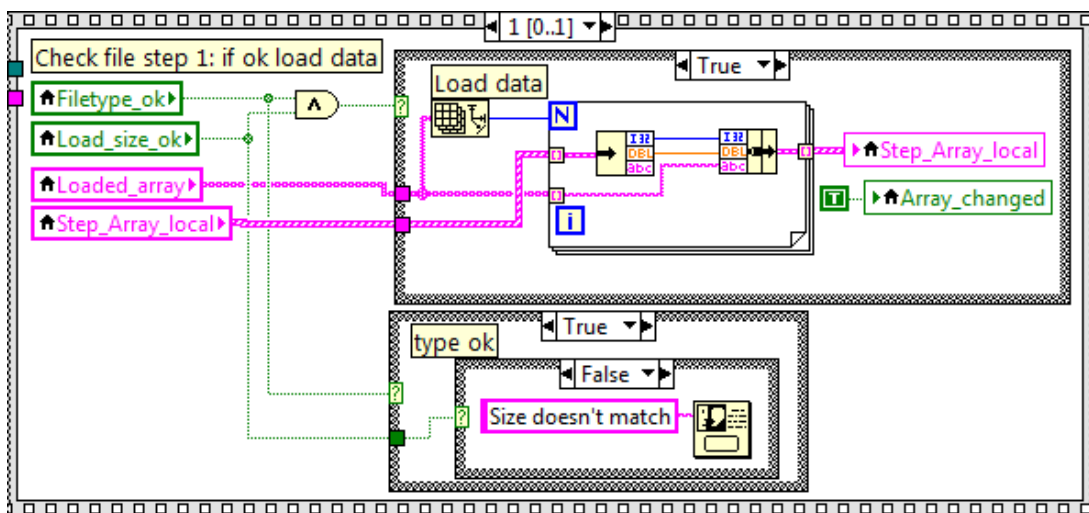
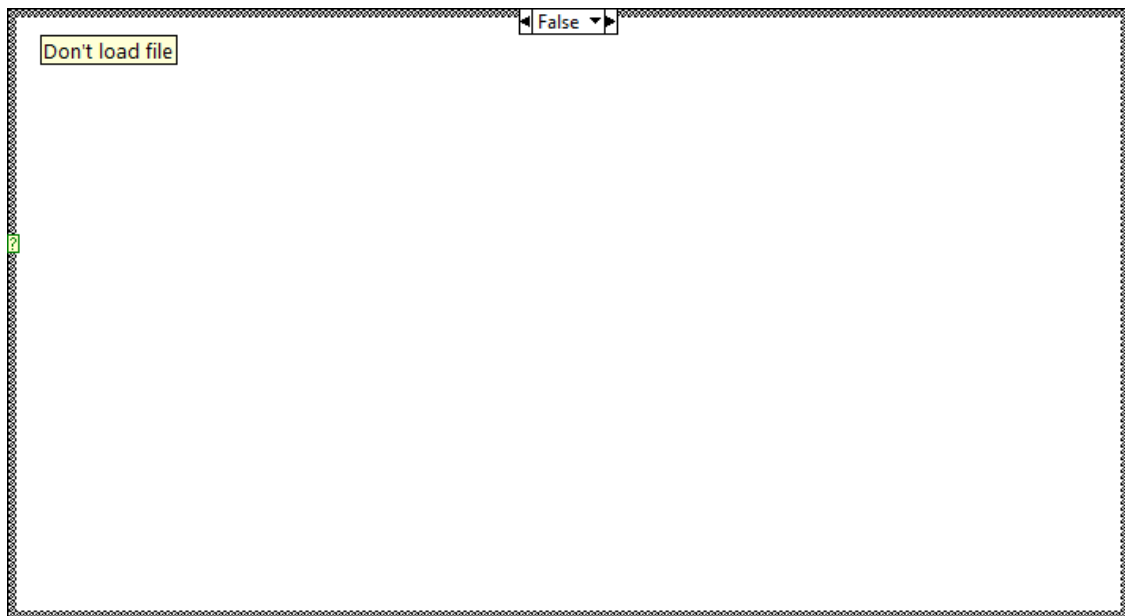


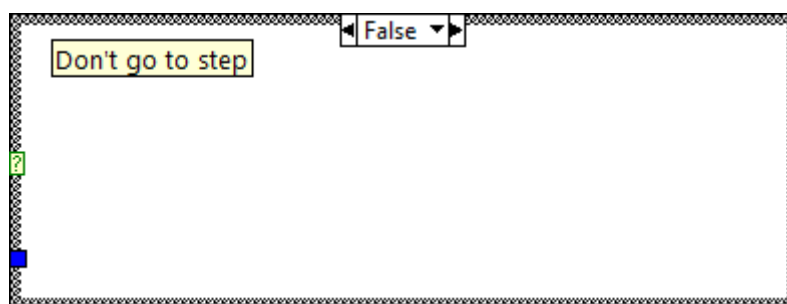
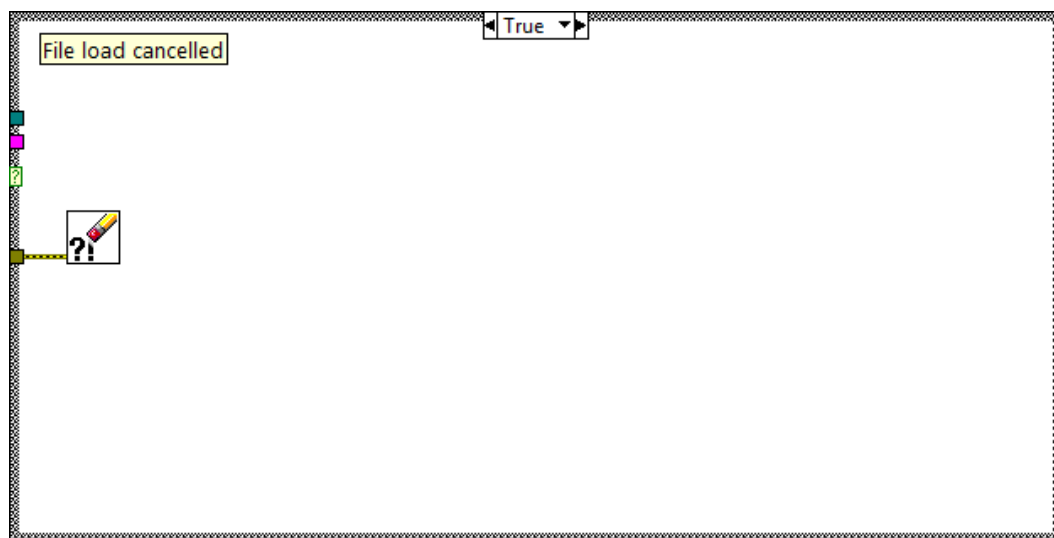


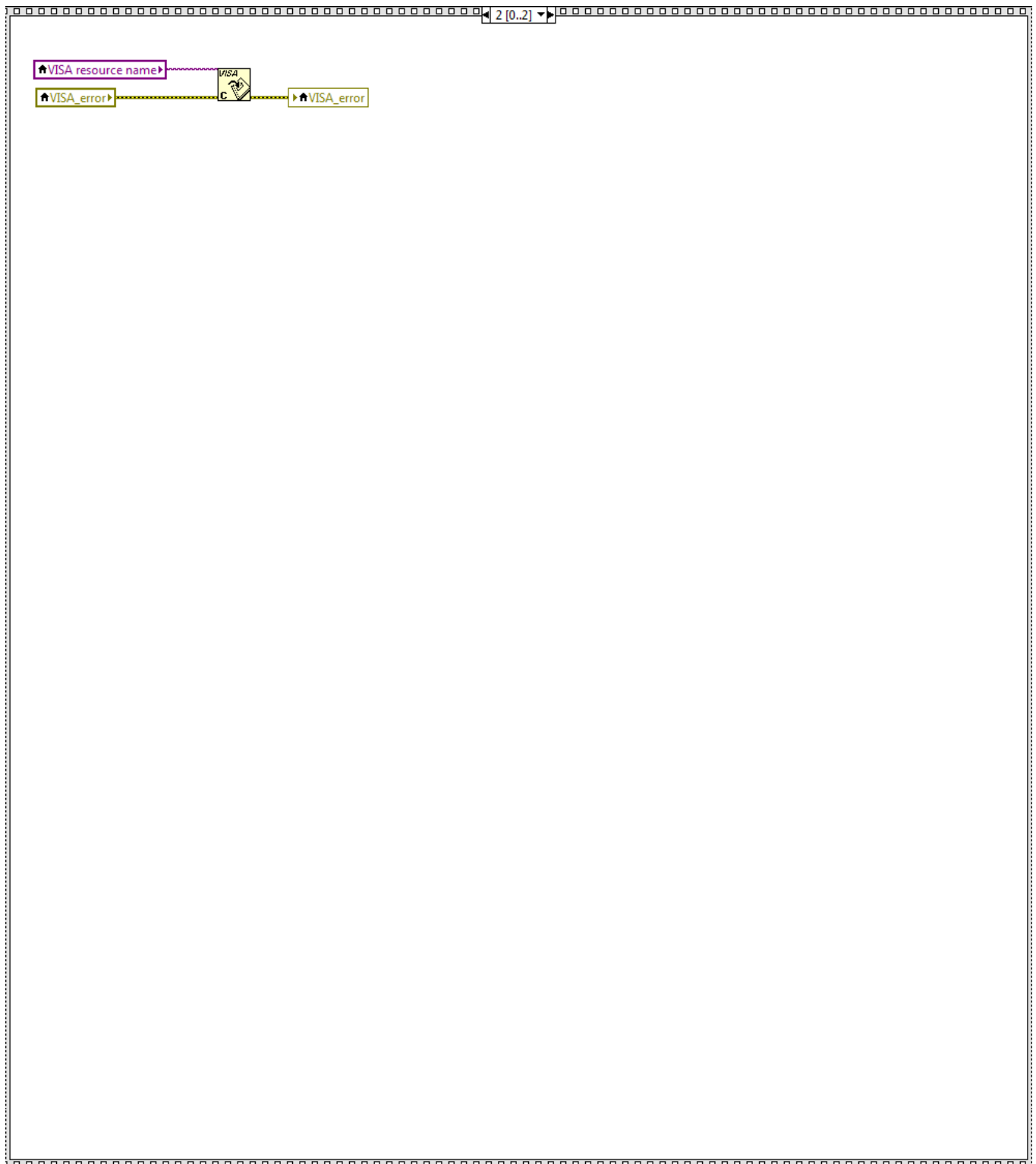






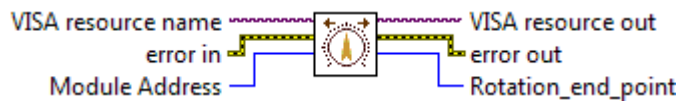




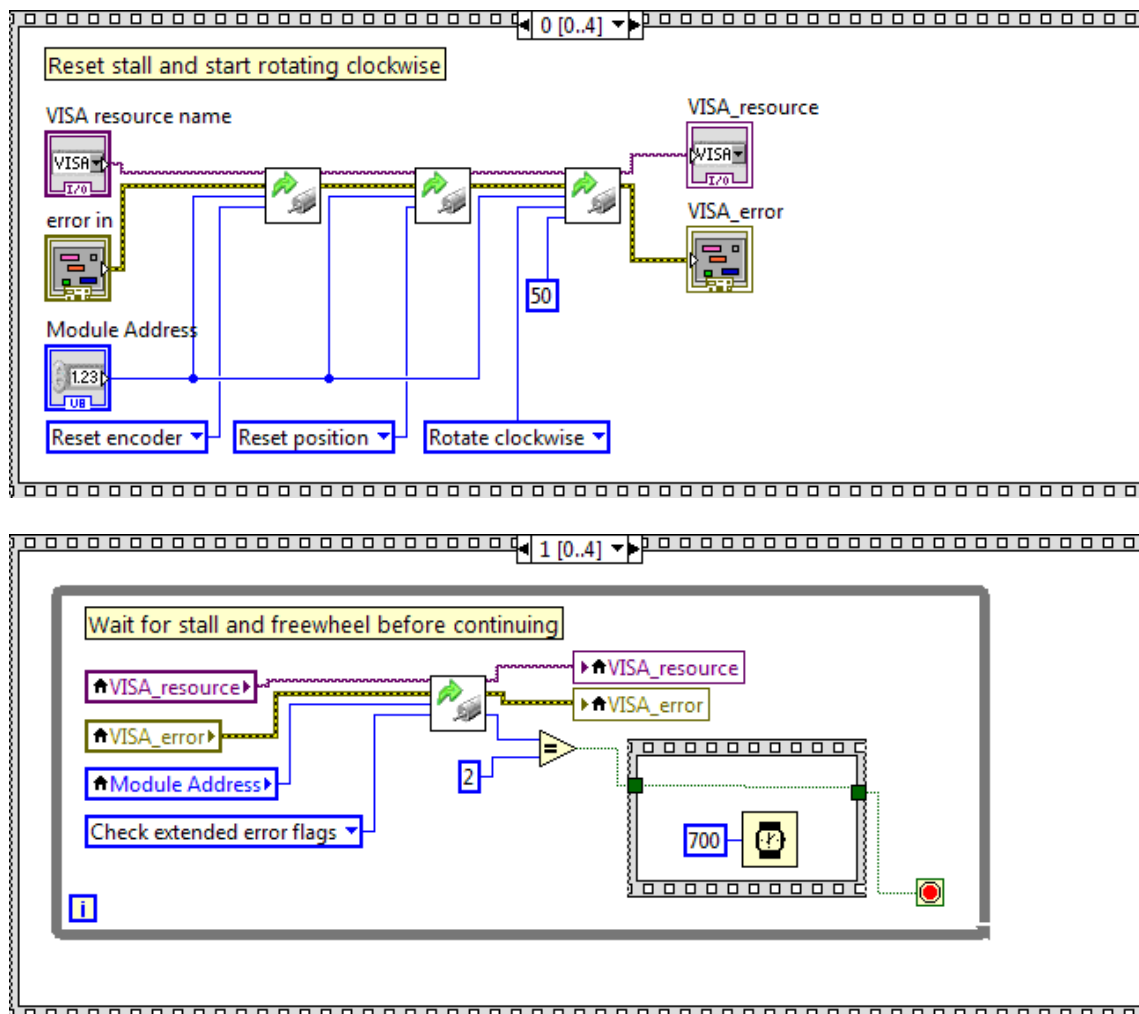


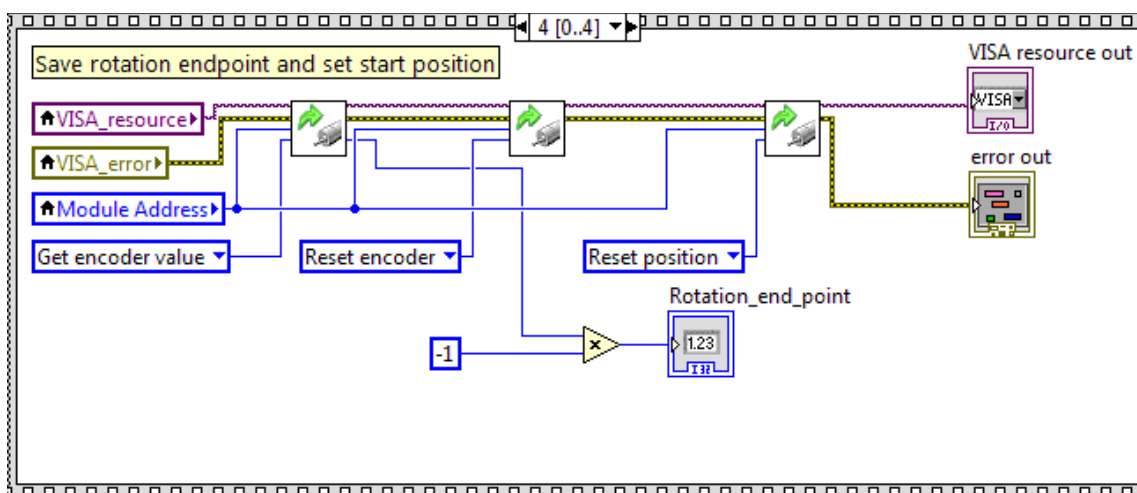
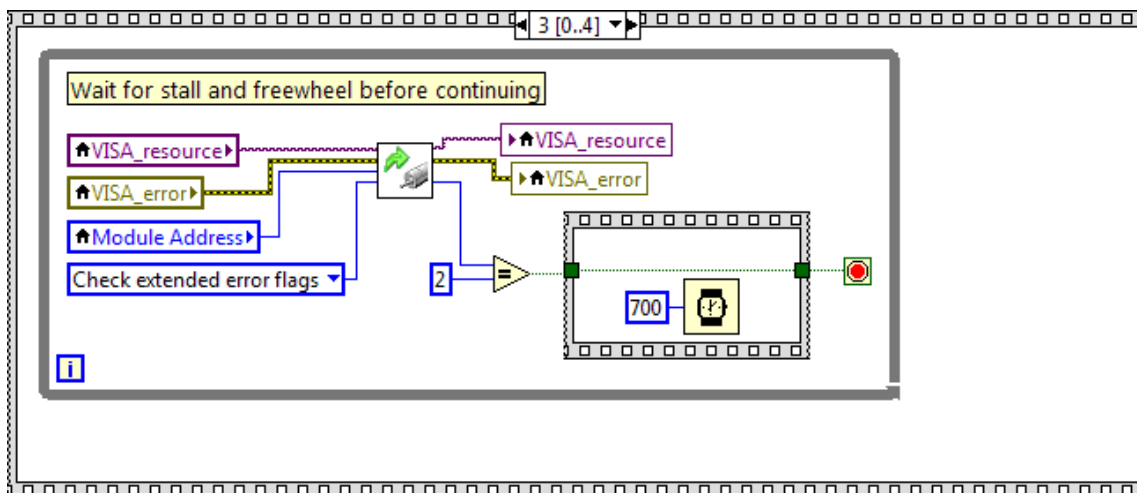
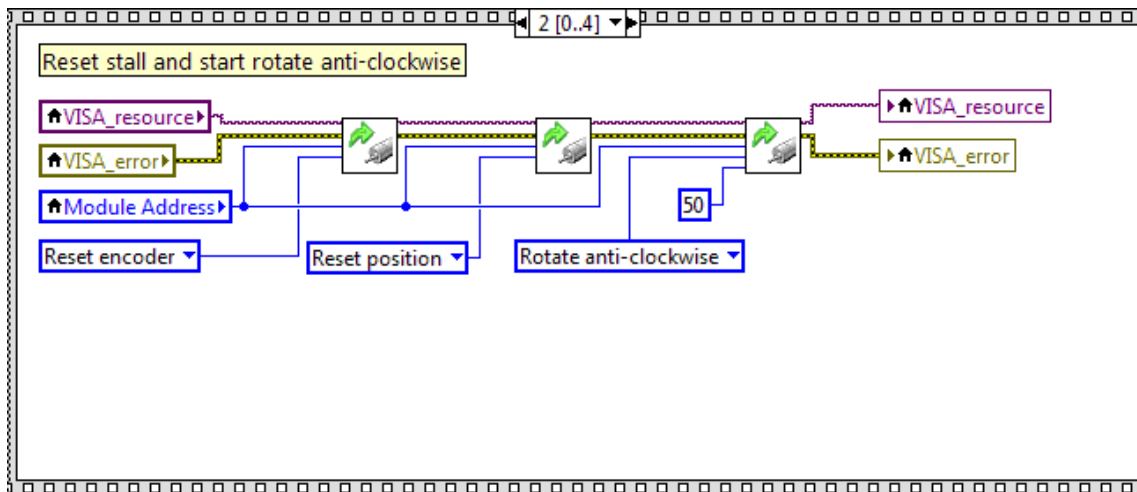
Find_endpoint.vi

Finds endpoints by rotating both directions until motor stalls and saves those positions.



Locating end points...



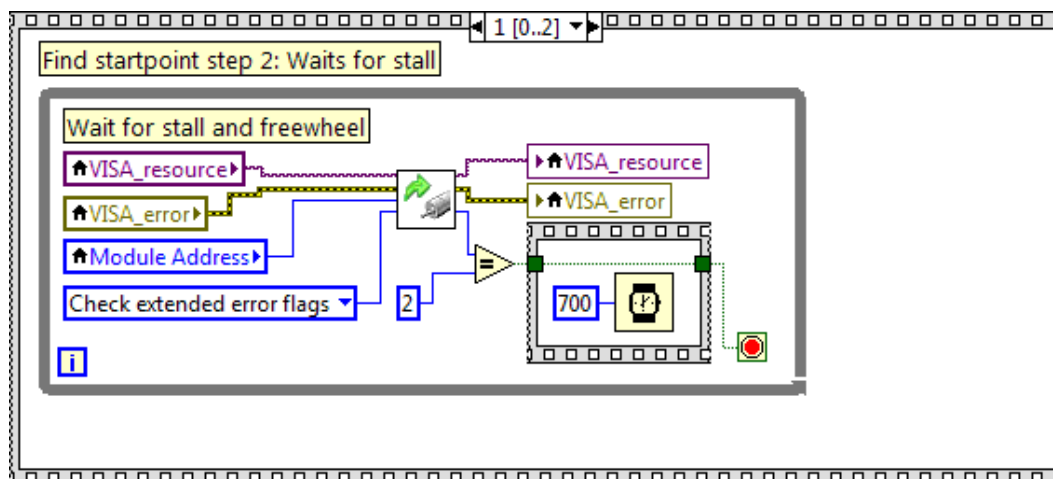
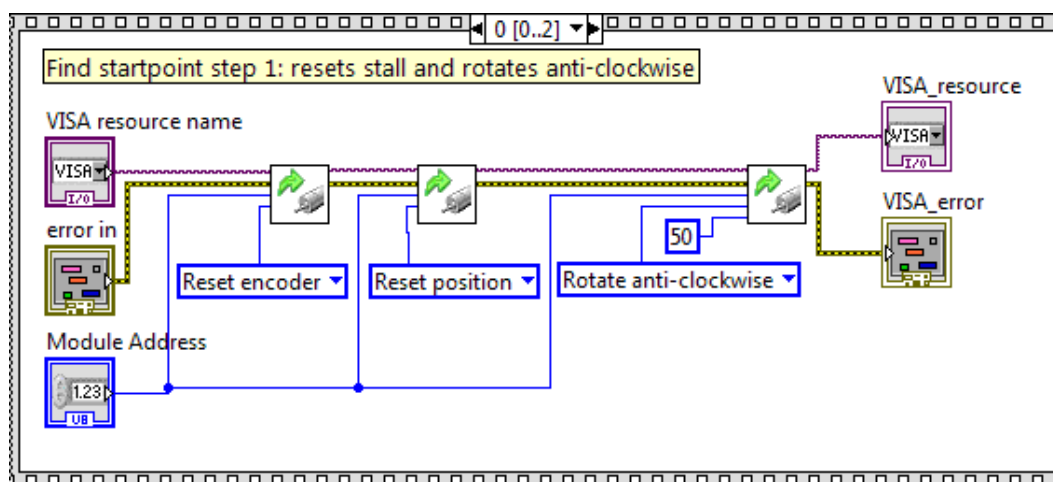


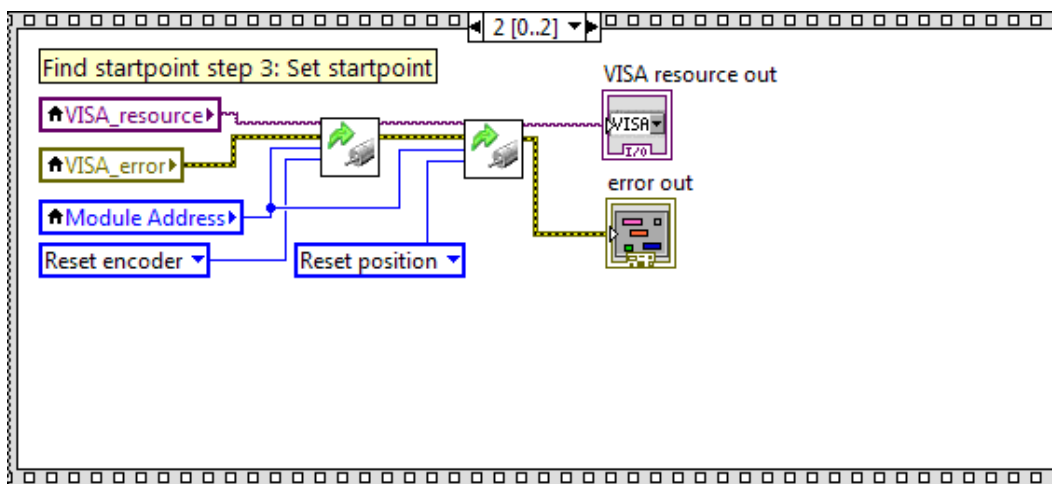
Find_startpoint.vi

Finds startpoint (zero degree position) of an end-to-end switch when a calibration file has been loaded.



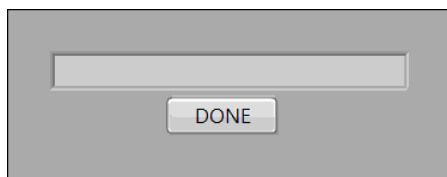
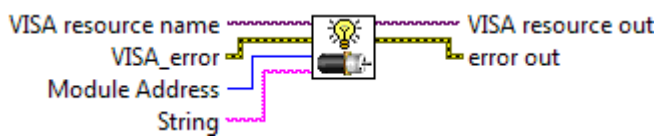
Locating start point...



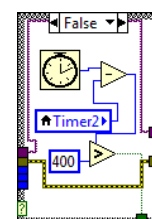
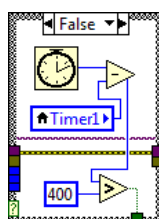
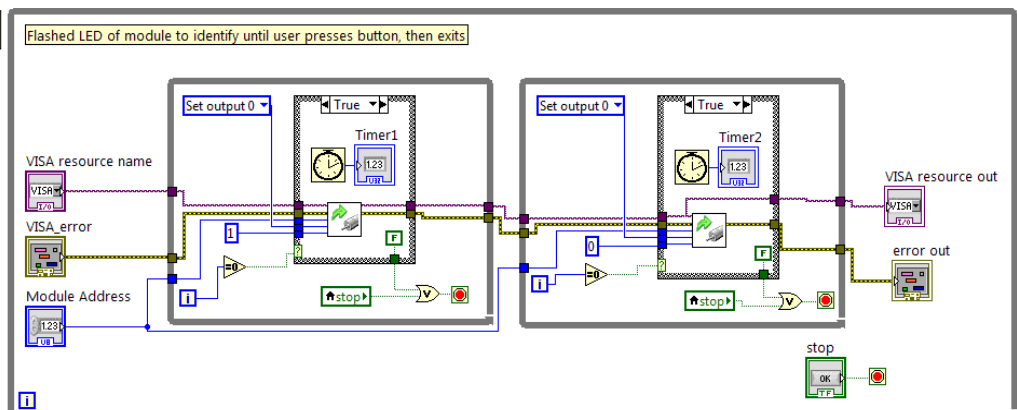
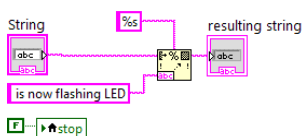


Identify_module.vi

Identifies a motor unit by flashing its LED.

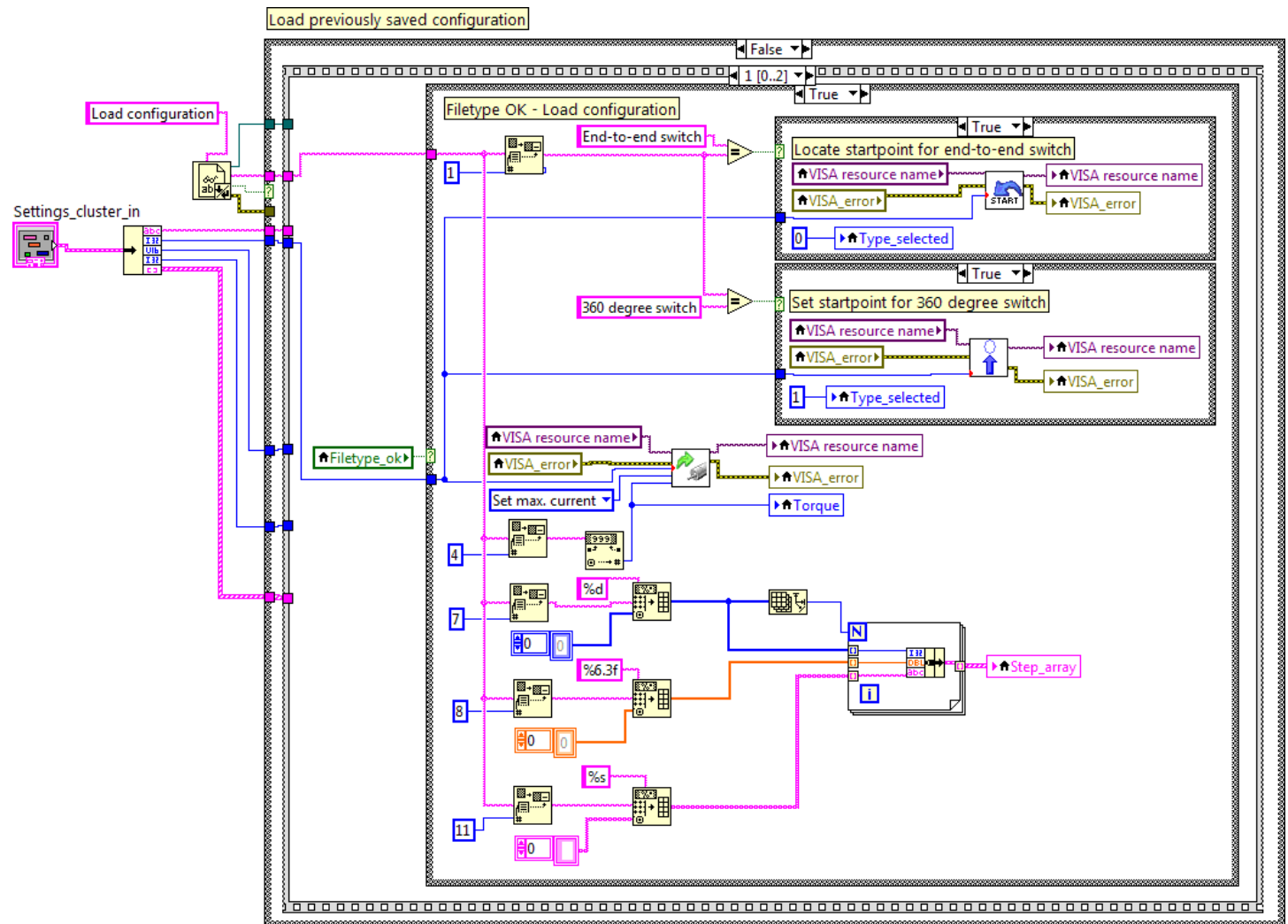


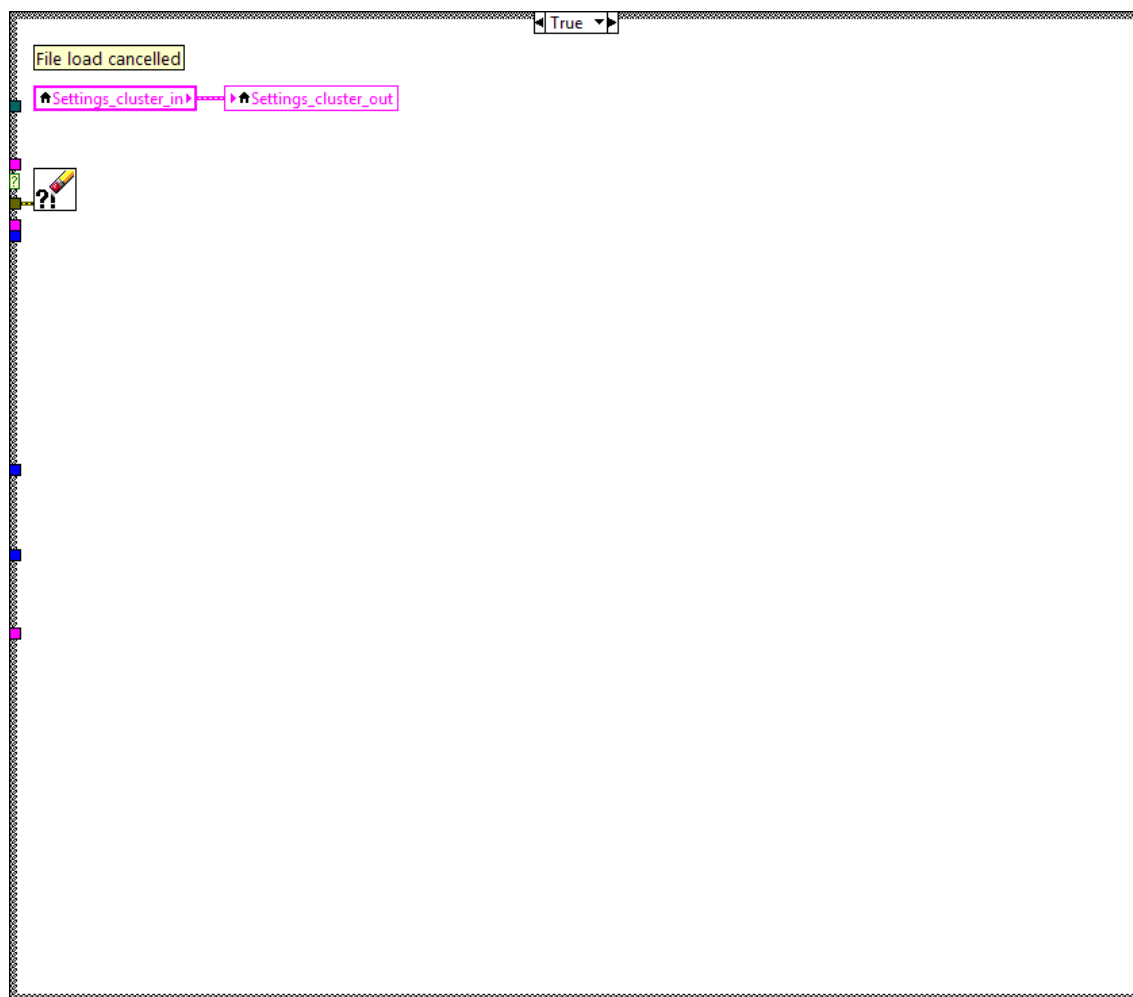
Generates status message containing the name of the module to identify

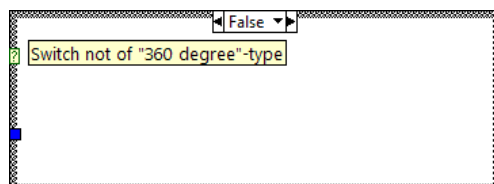
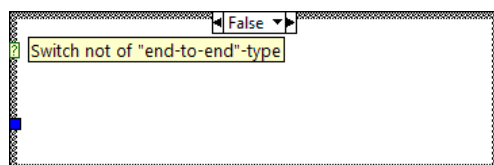
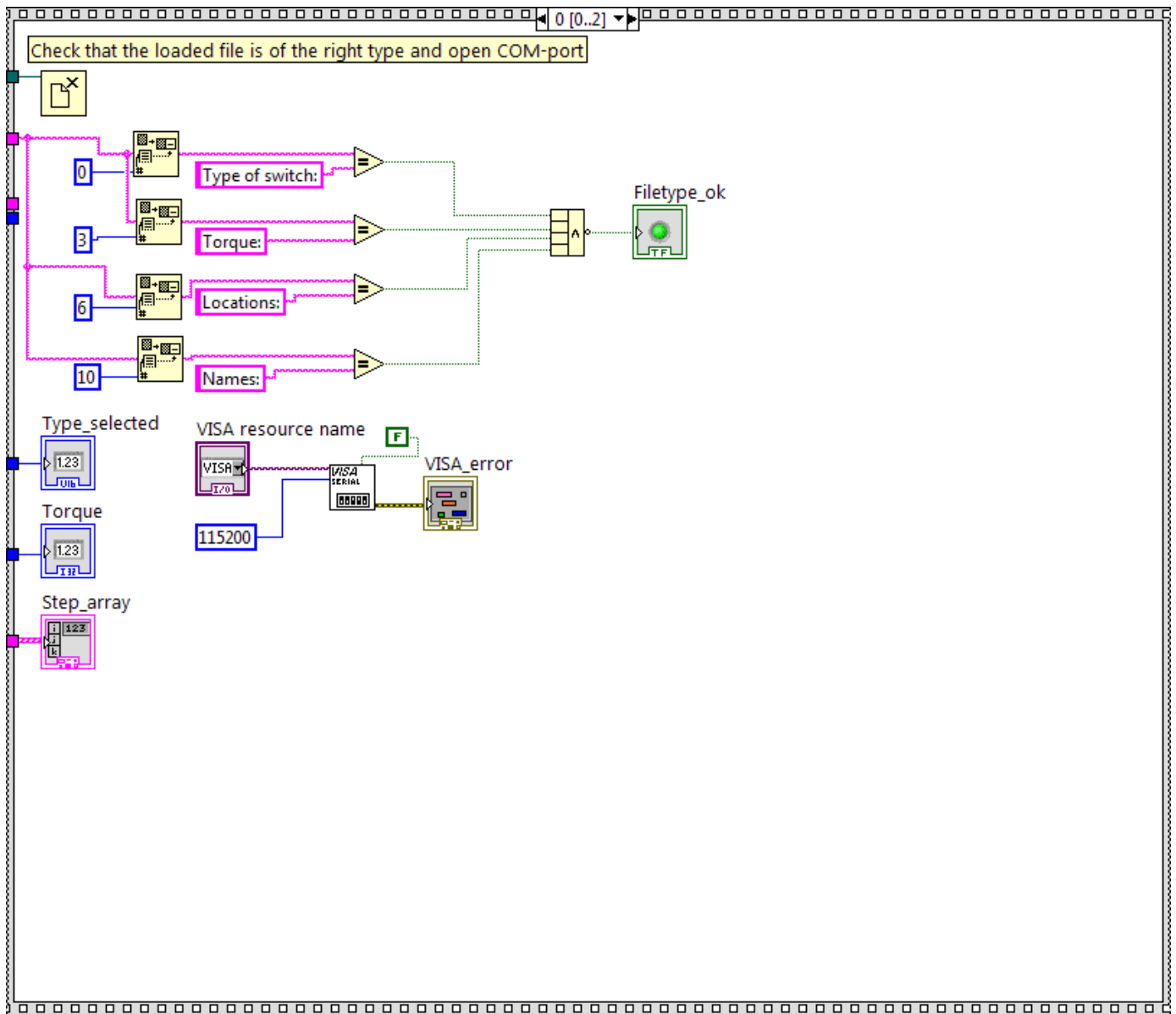


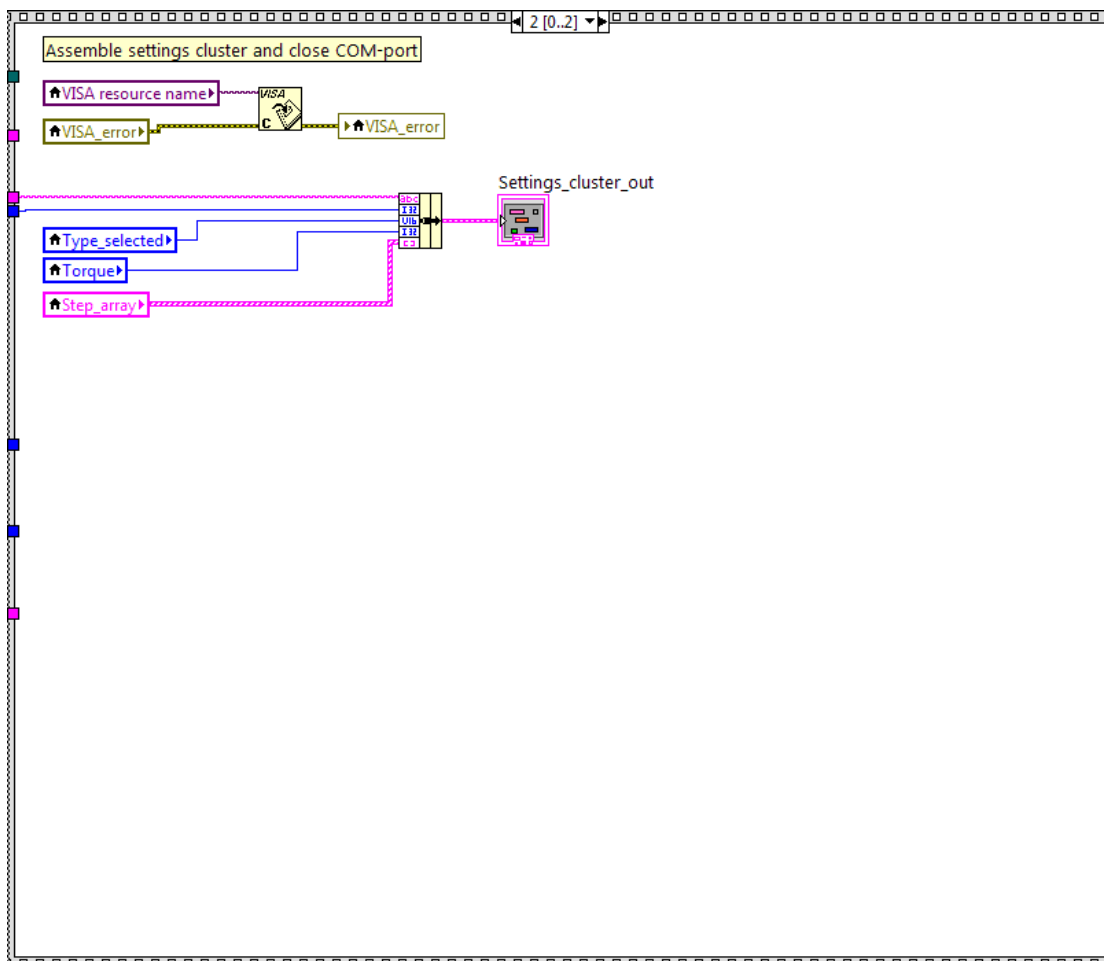
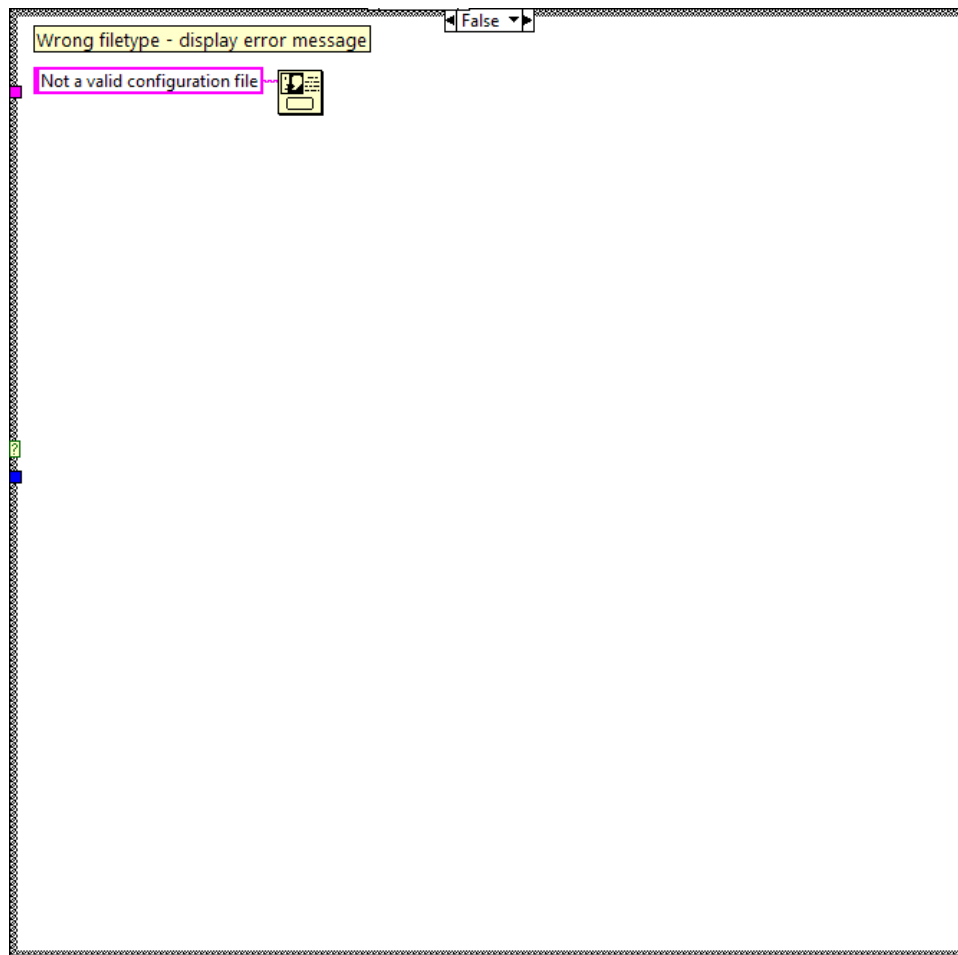
load_config_file.vi

Loads previously saved calibration file and locates the start point of the switch.



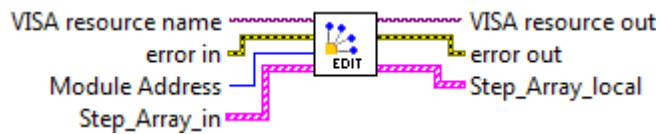






Manual_step_adder.vi

Used to manually calibrate steps in a switch. Allows user to add step positions one at a time.



Manual Step Adder

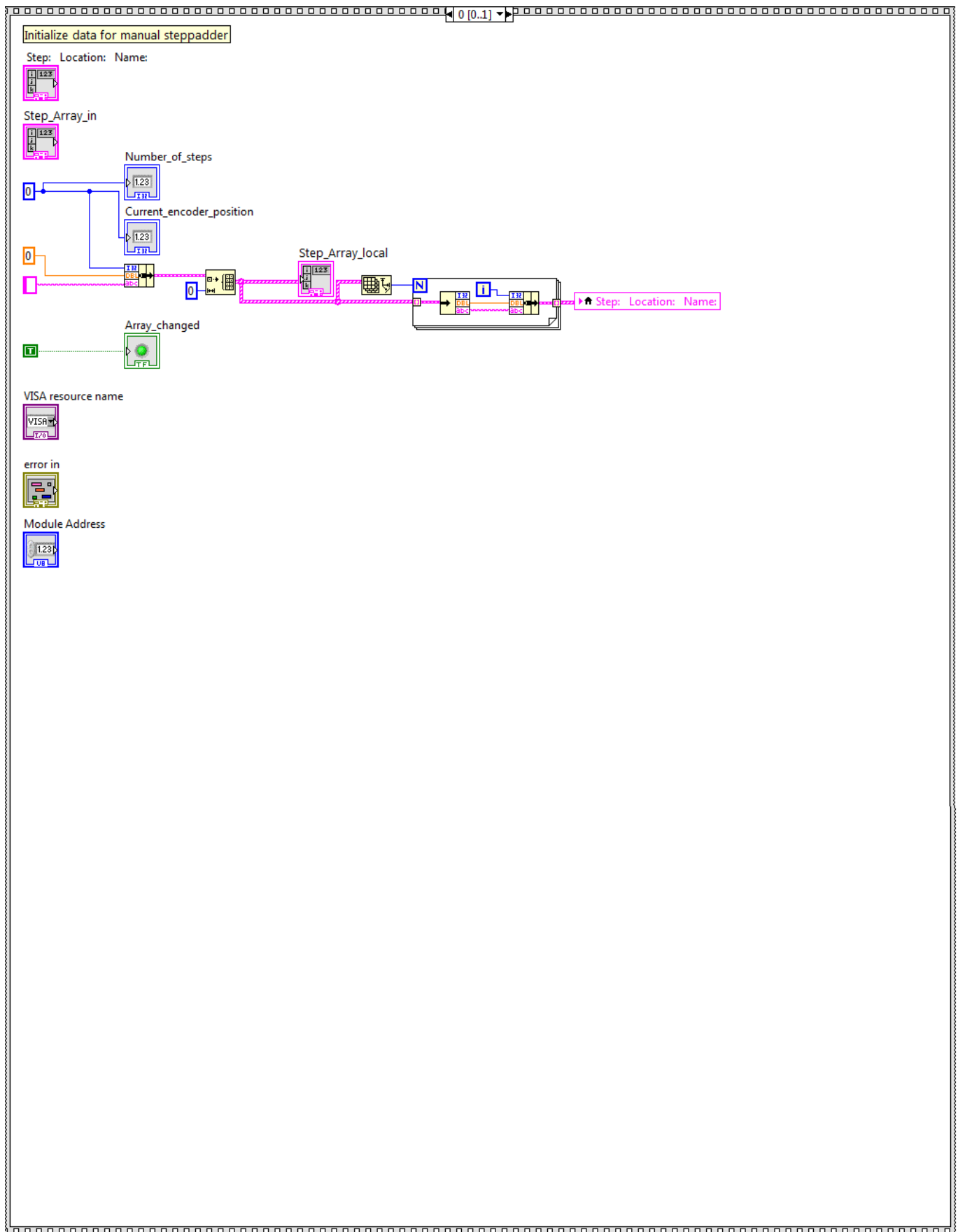
Step: Location: Name:

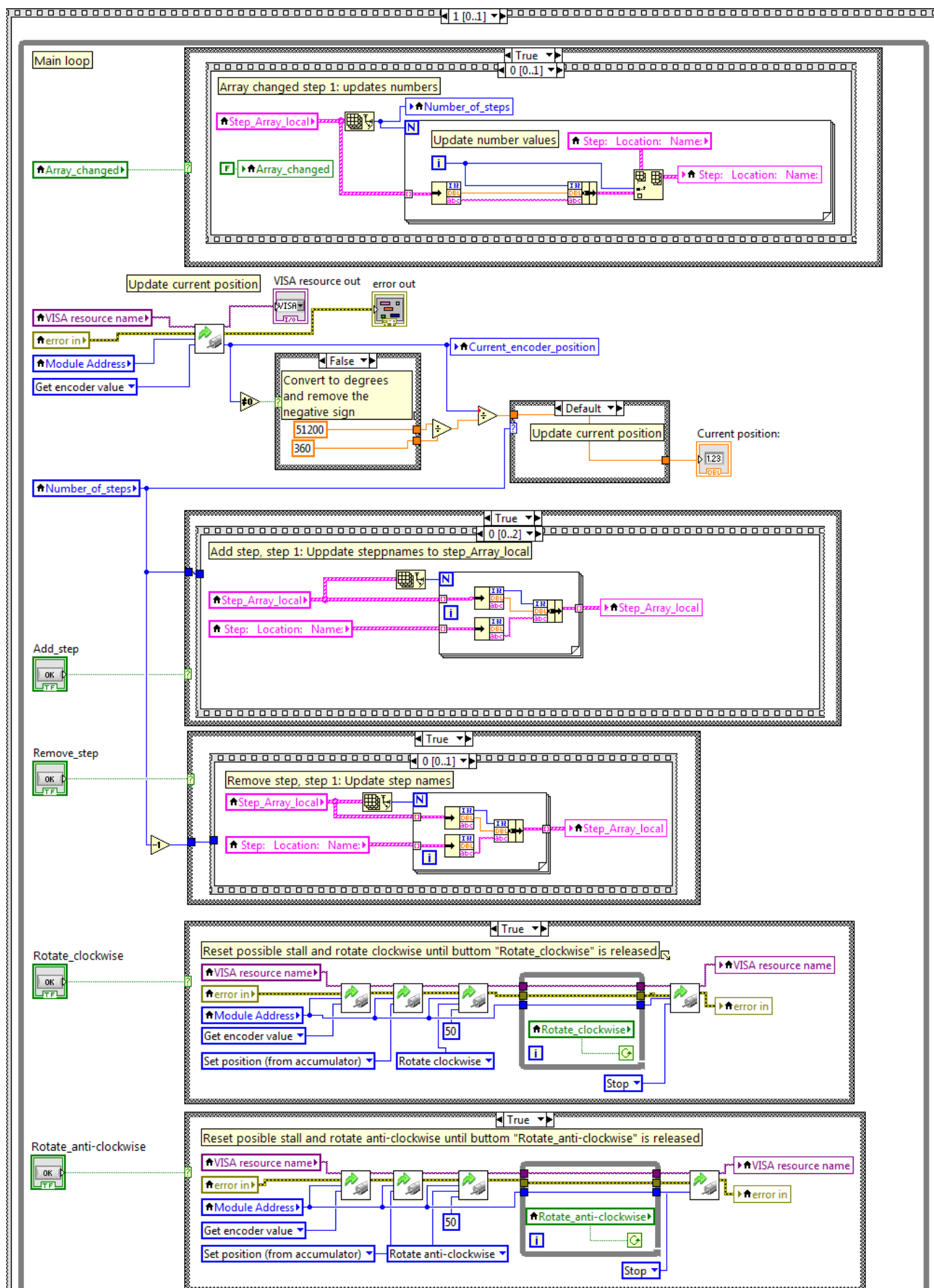
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	
0	0	

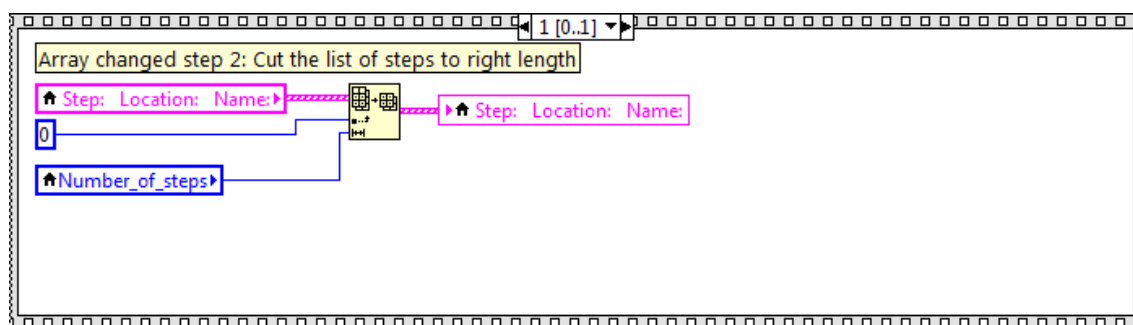
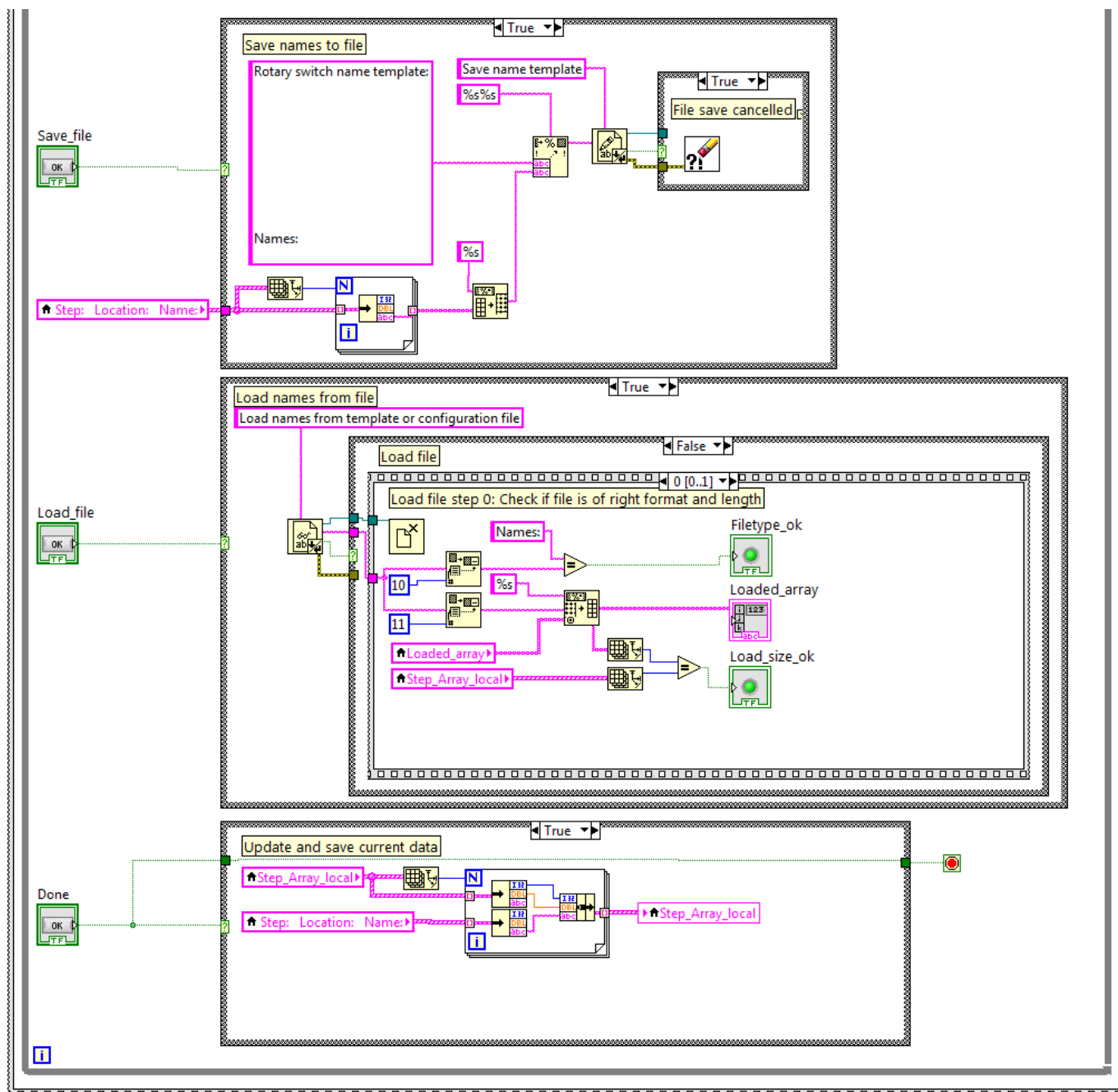
ADD STEP POSITIONS

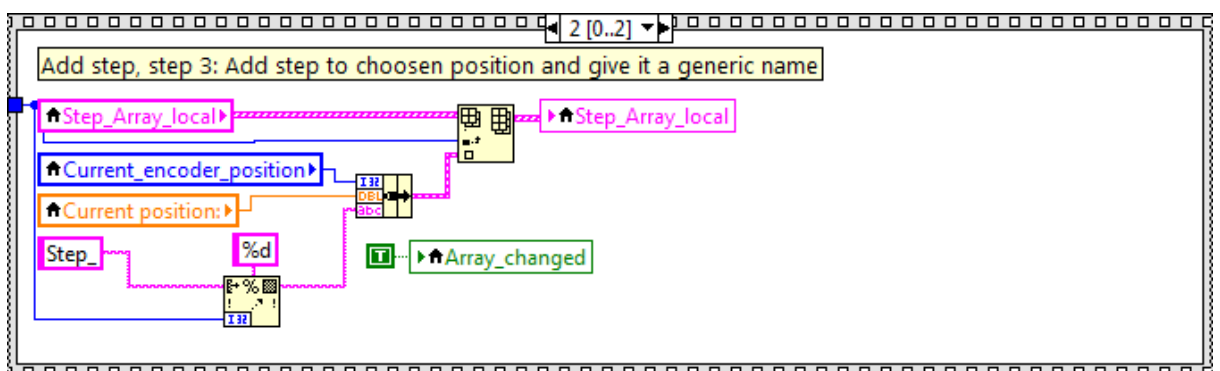
Current position:

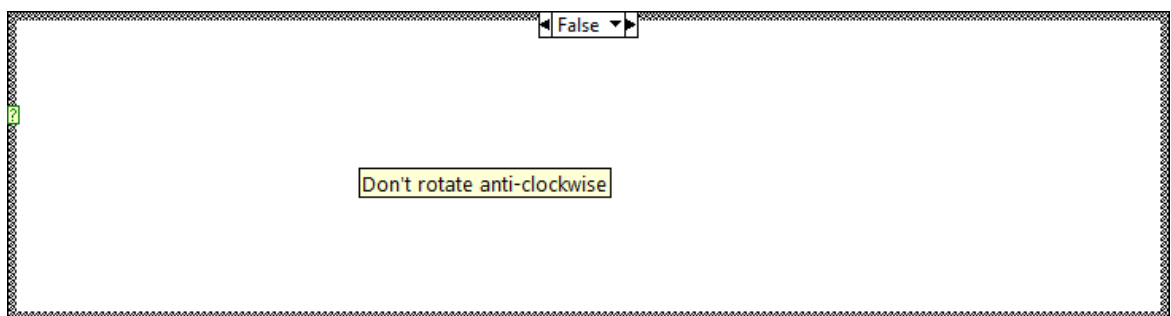
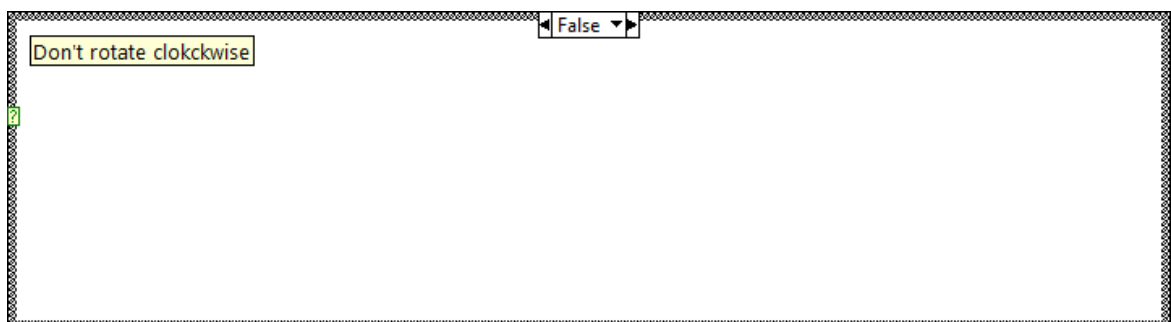
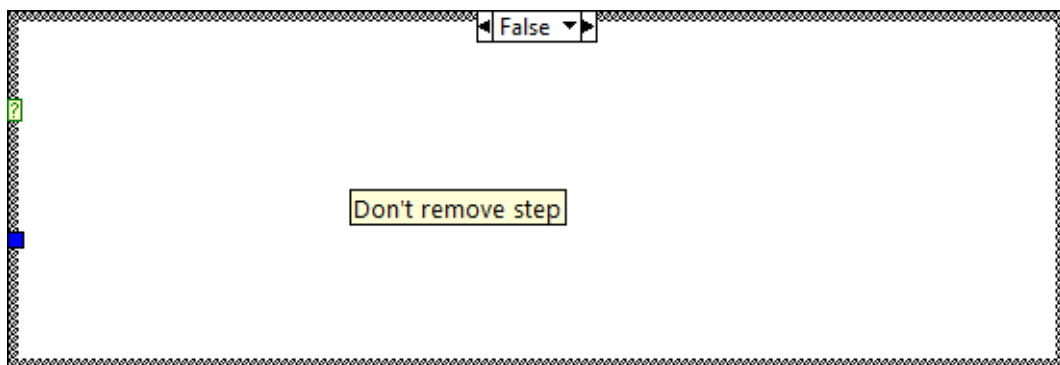
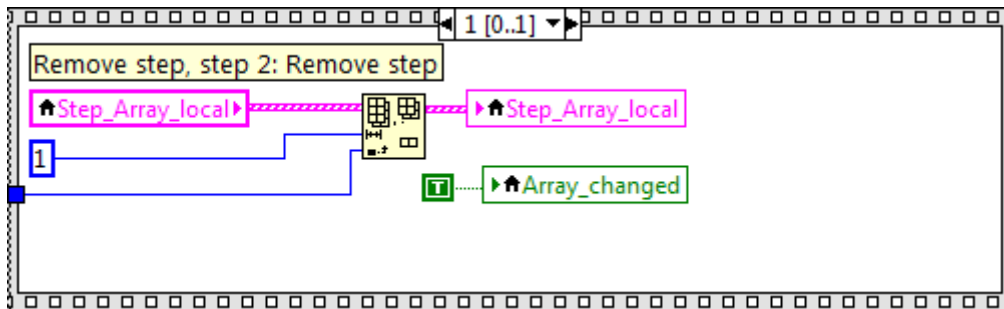
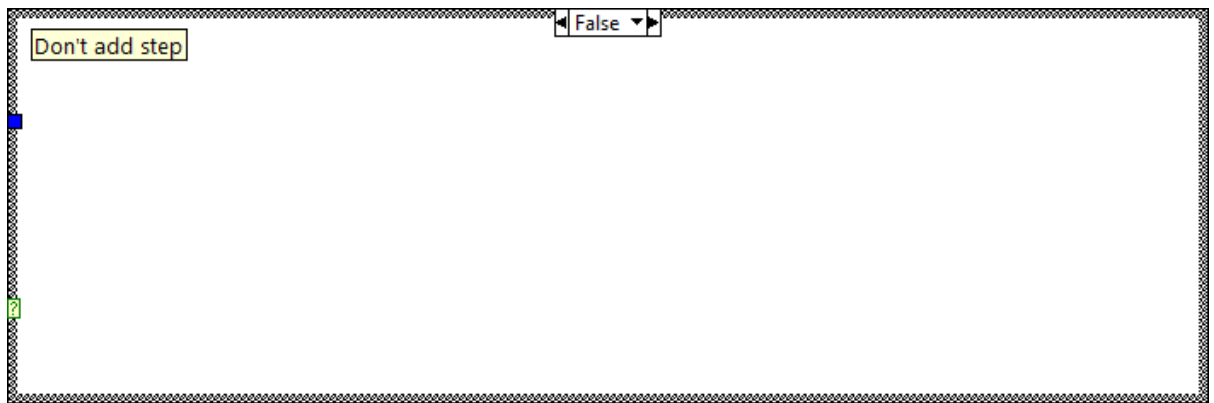
SAVE/LOAD NAME TEMPLATE

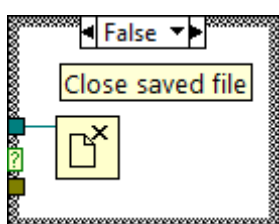
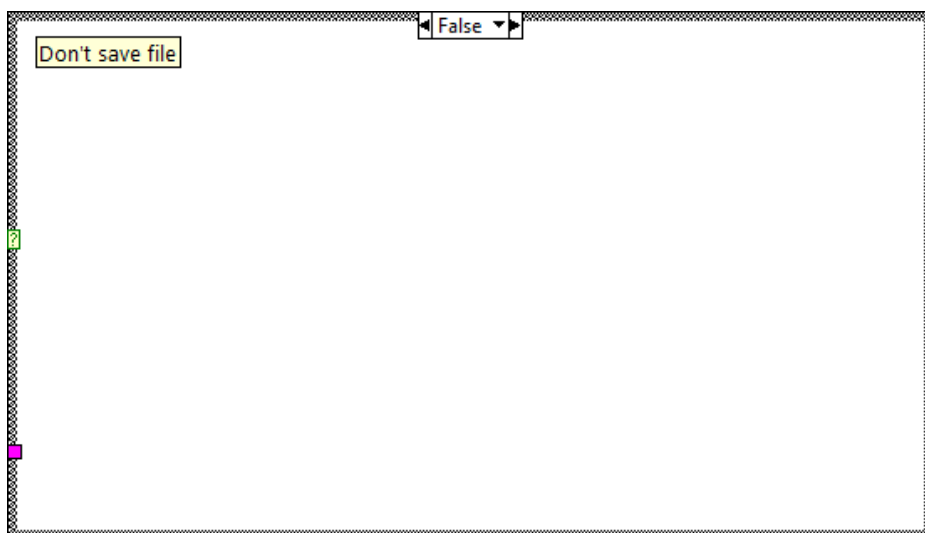


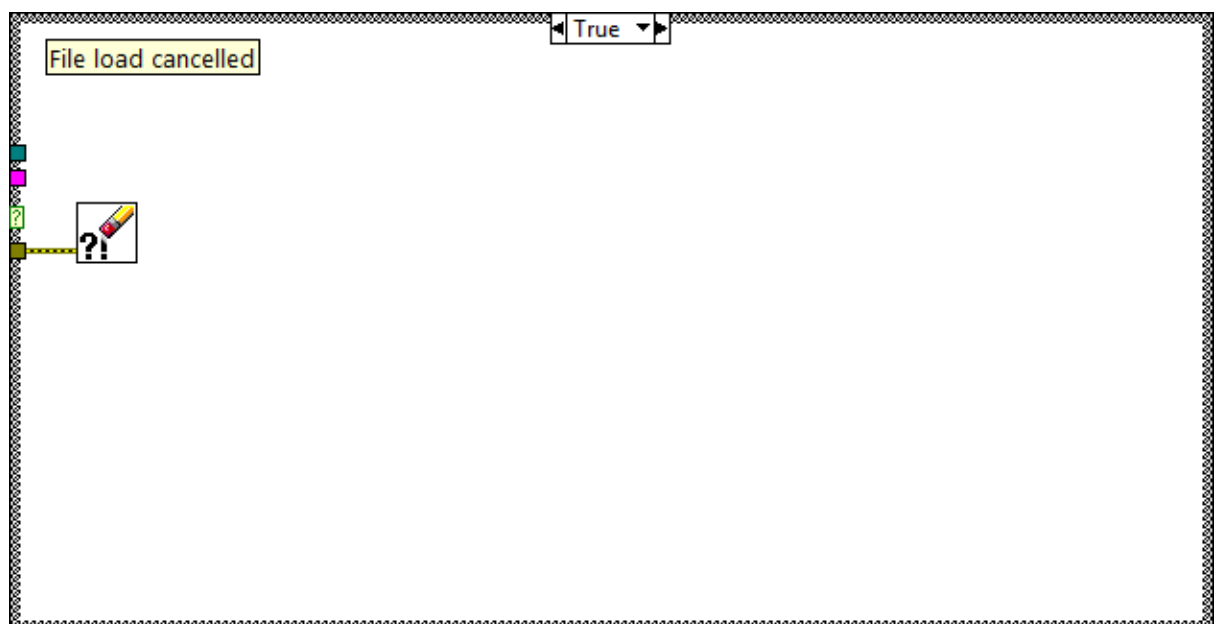


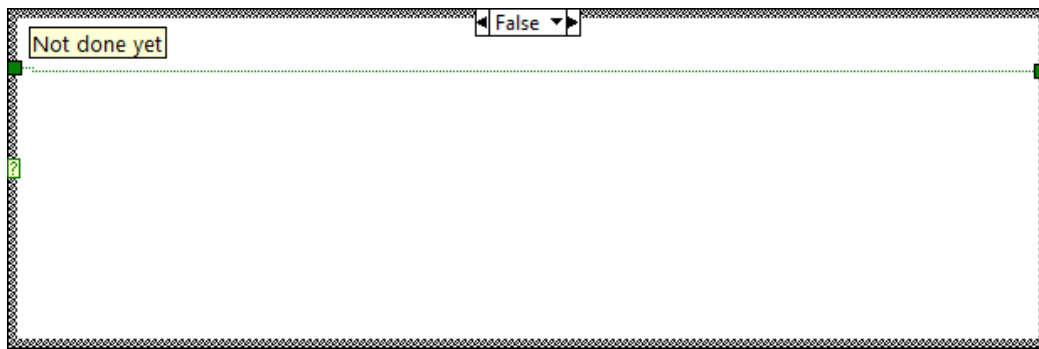






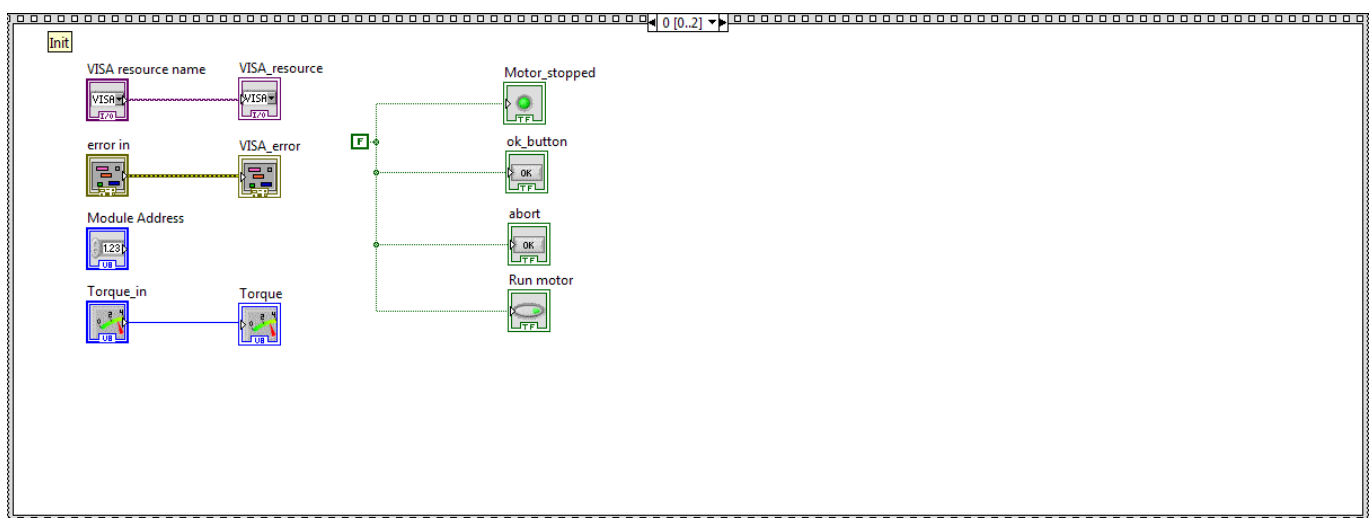
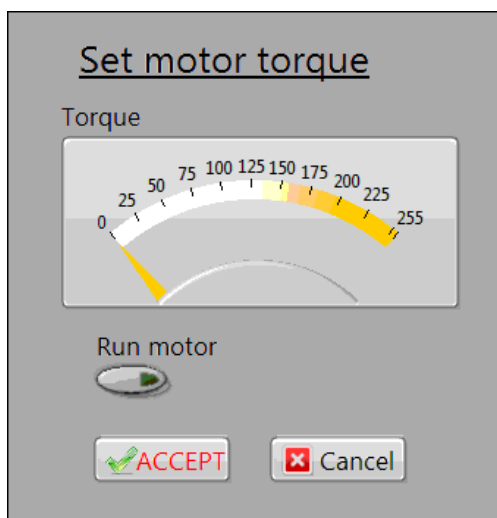
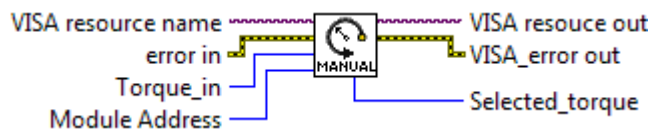


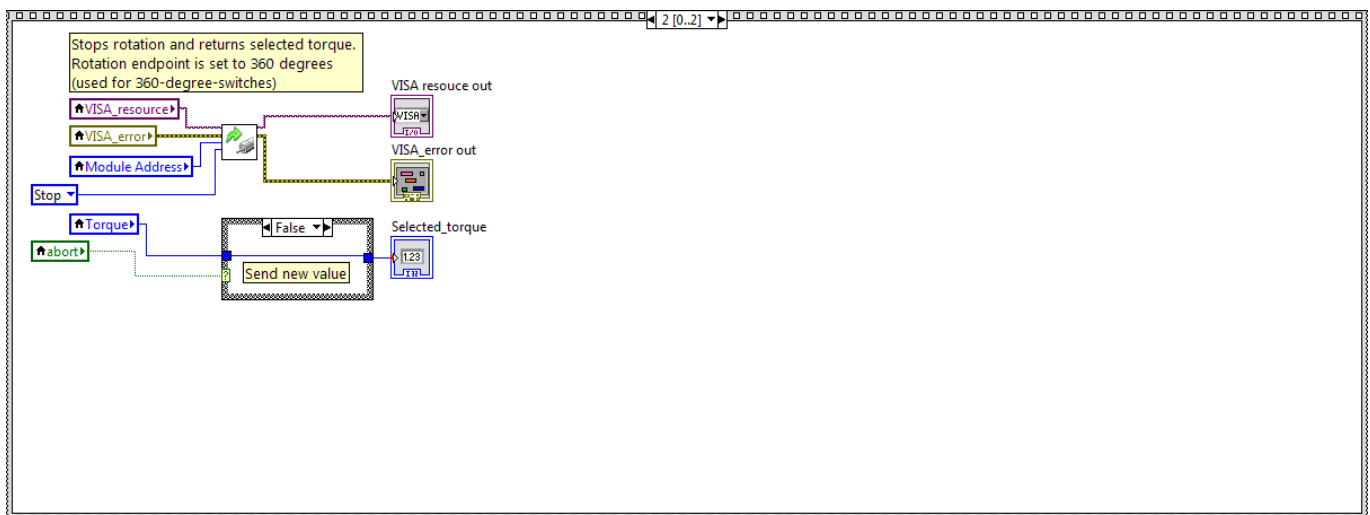


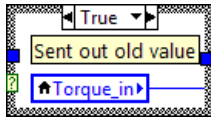


Manual_torque_calibrate.vi

Used to manually configure motor torque.

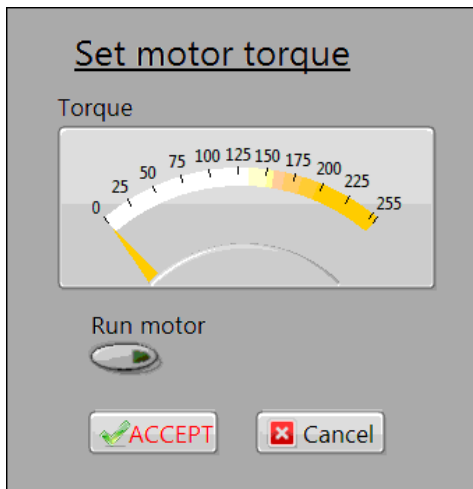


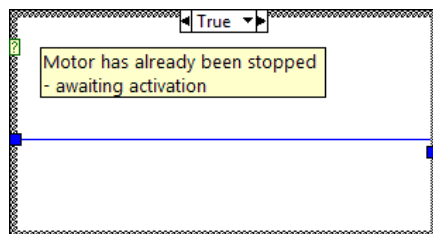
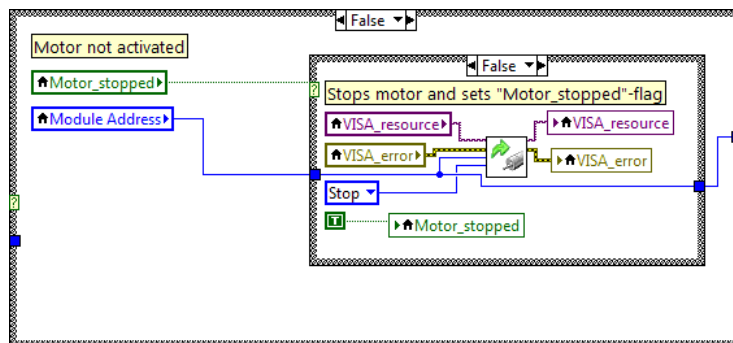
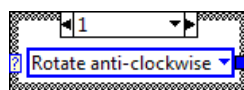
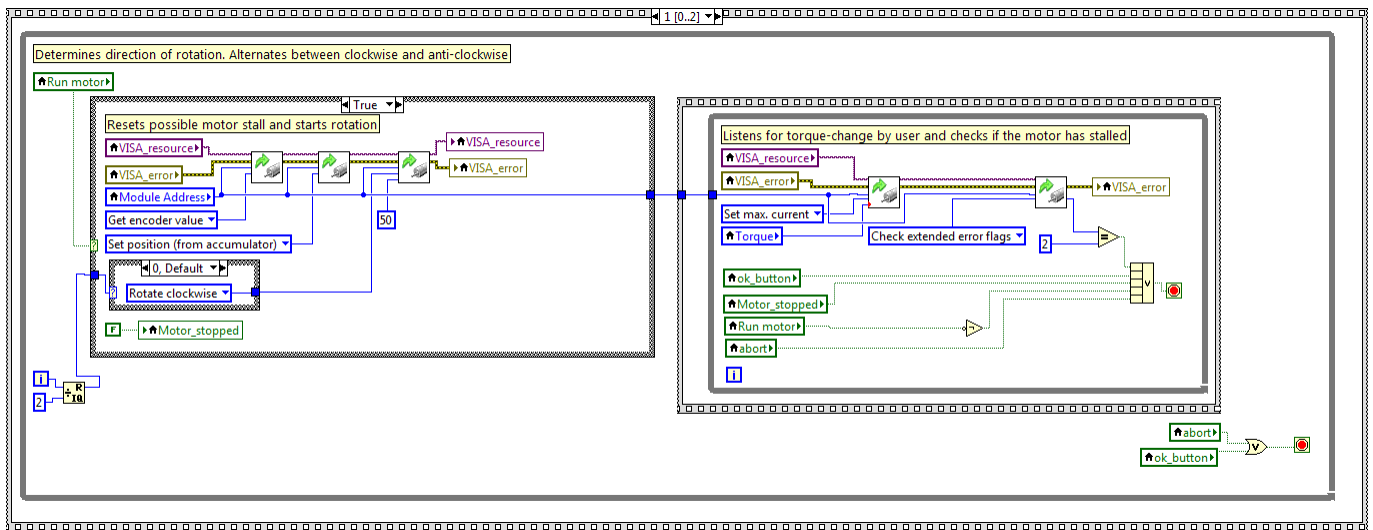


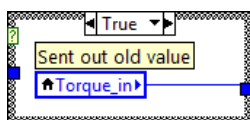


Manual_torque_edit.vi

Used to edit torque in an existing calibration.

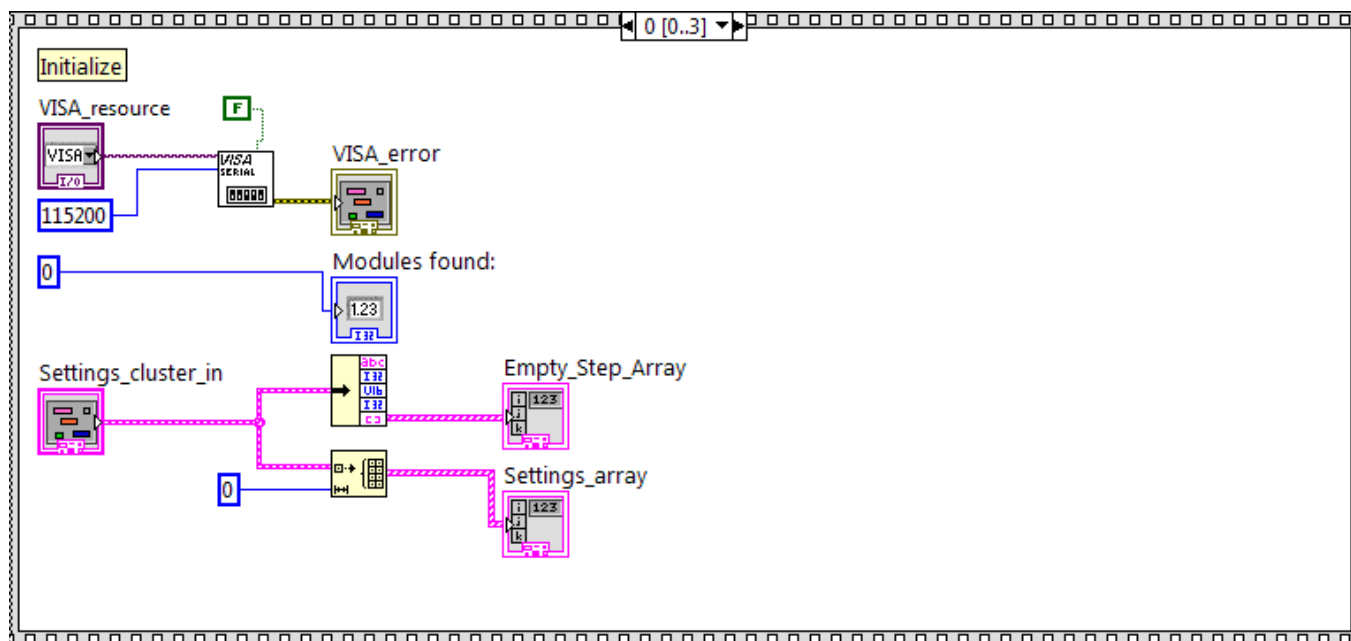
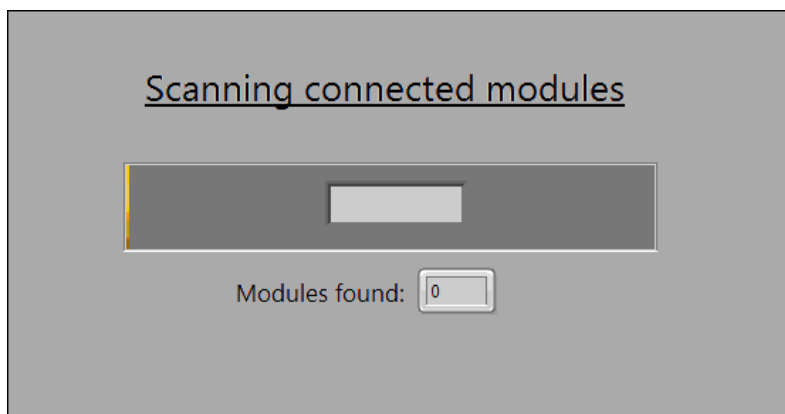


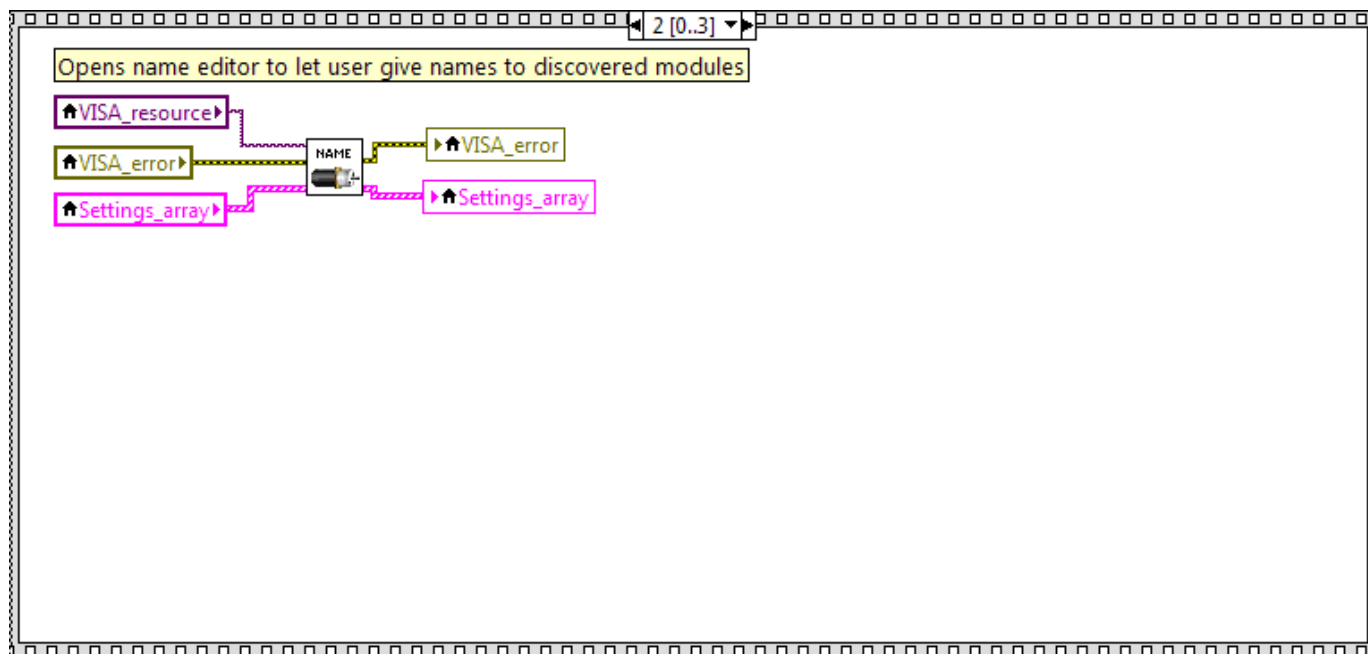
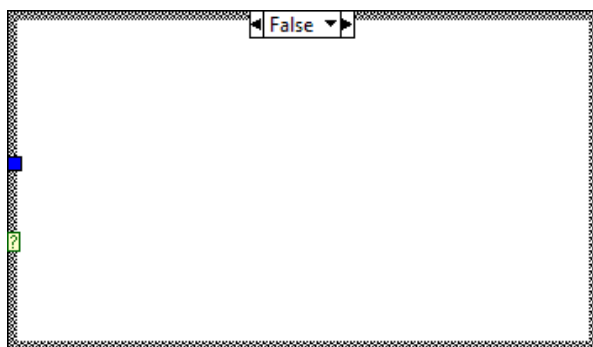
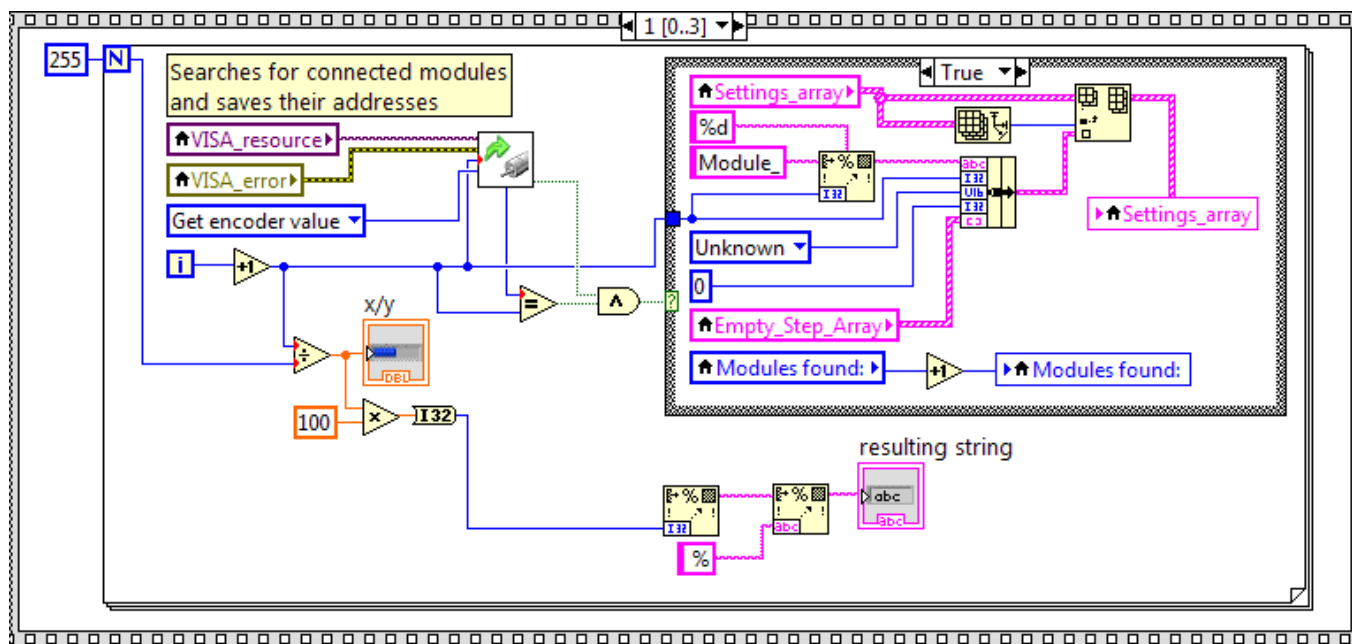


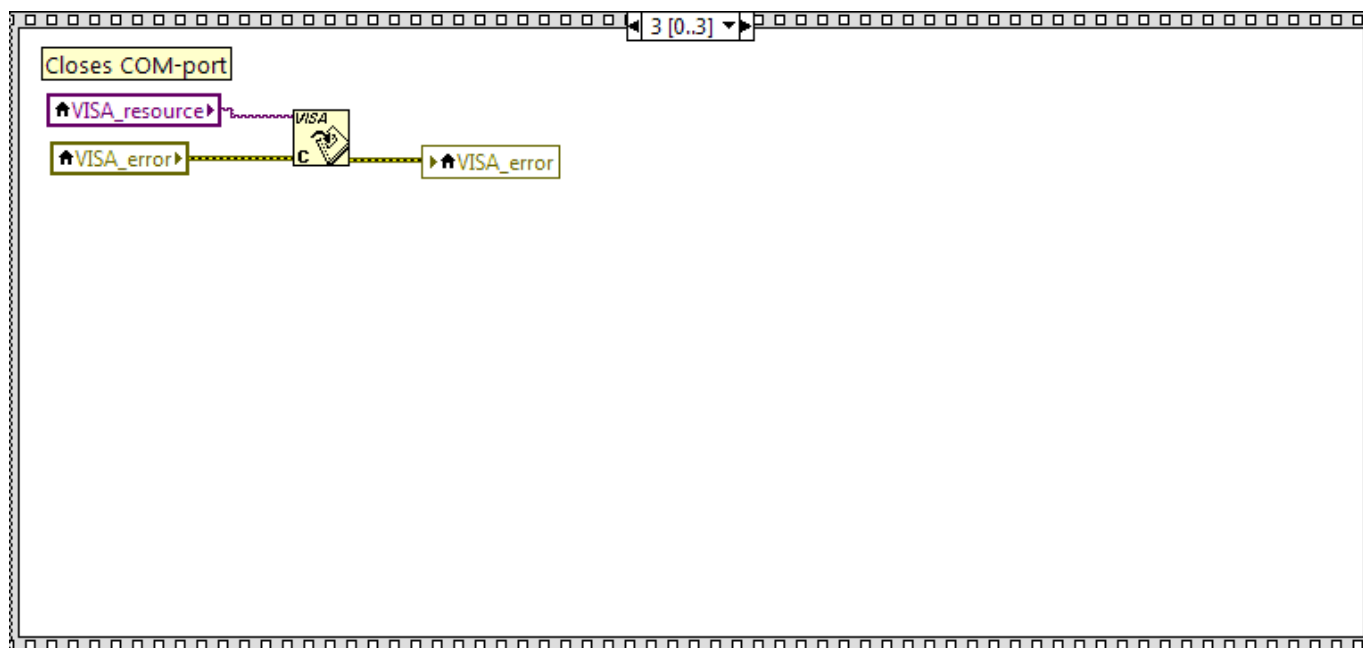


Module_scanner.vi

Scans for connected motor units to calibrate.







Module_scanner_name_editor.vi

Lets user give names to connected motor units.

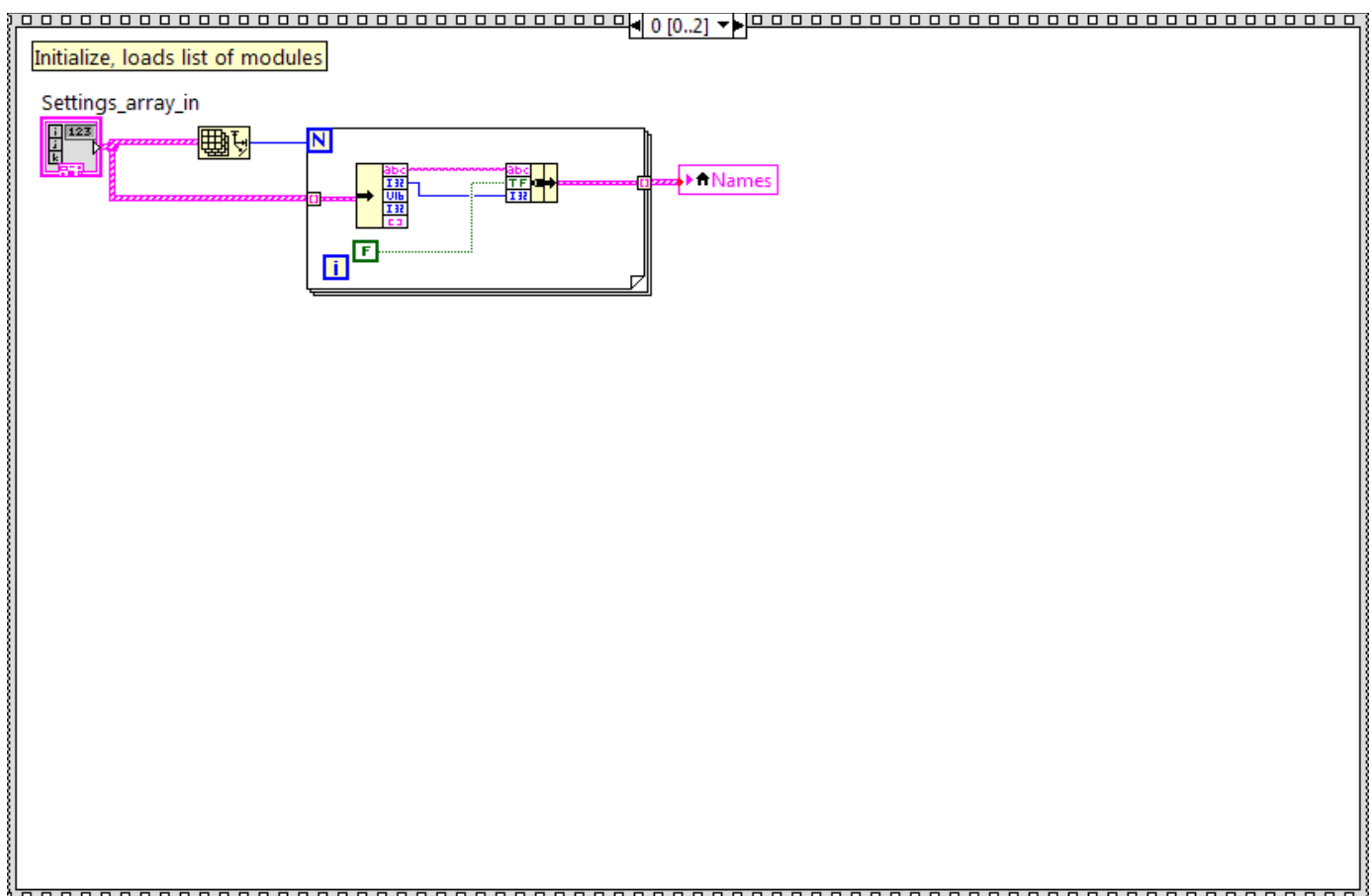


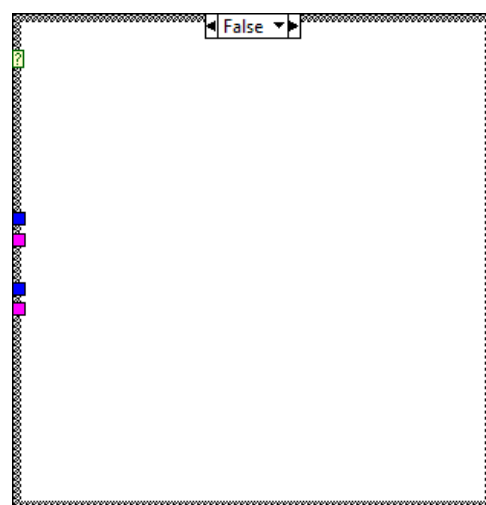
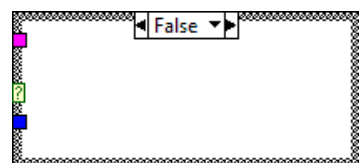
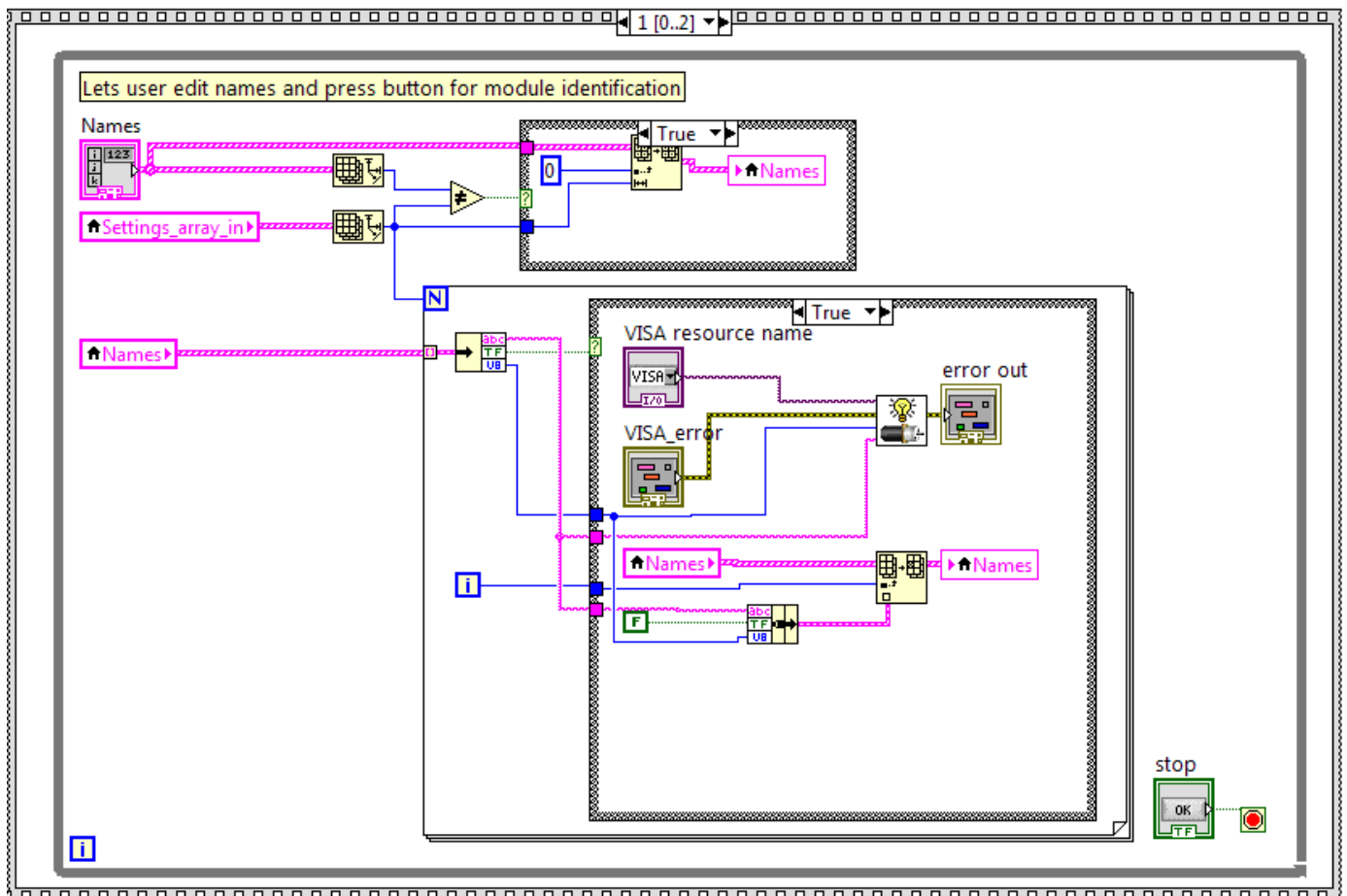
Set Module names

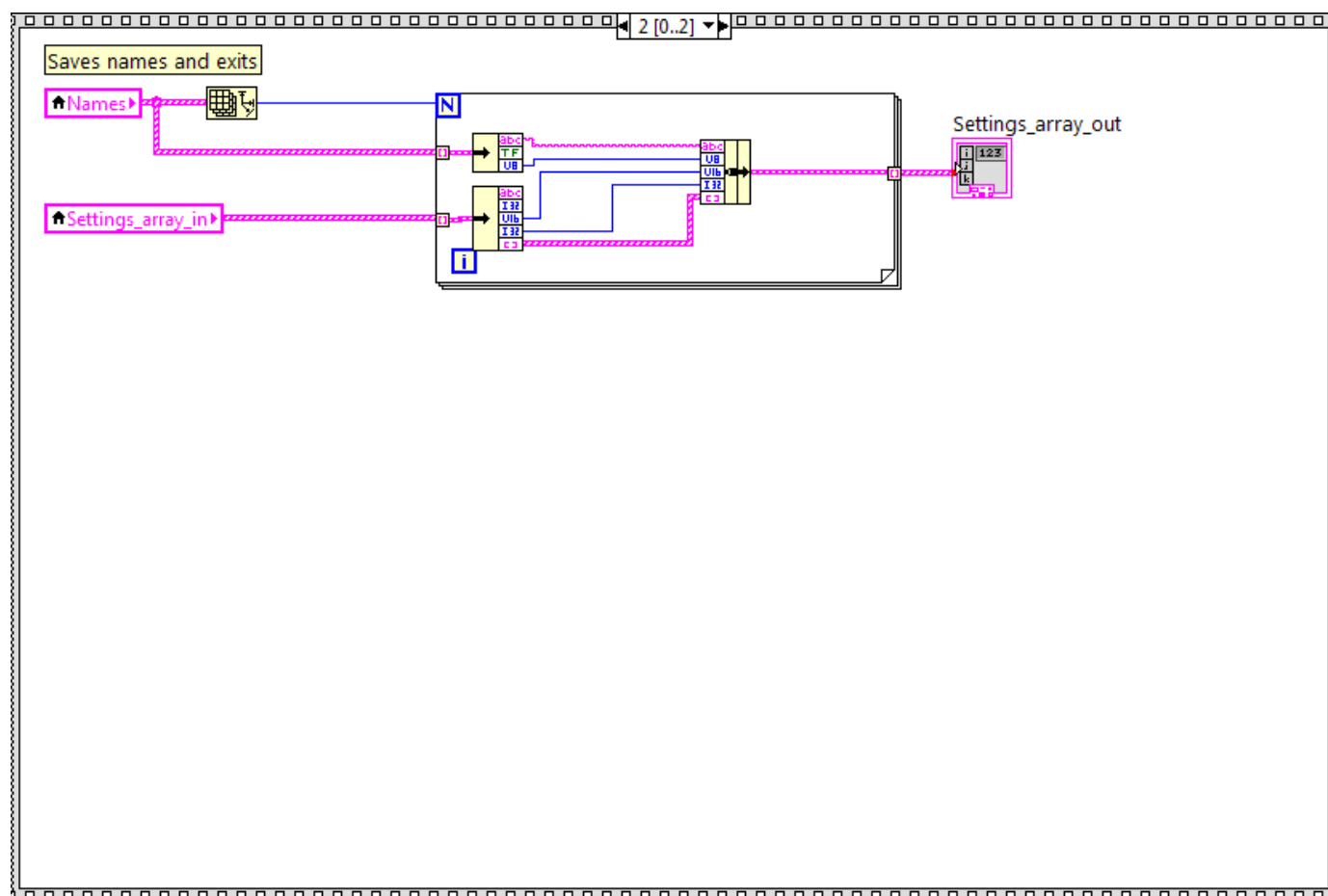
Address: Name:

0	Identify	
0	Identify	
0	Identify	
0	Identify	
0	Identify	
0	Identify	
0	Identify	

 ACCEPT

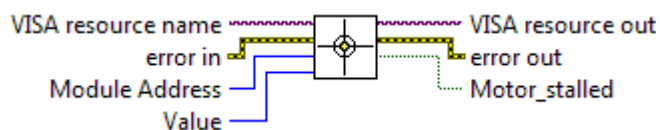




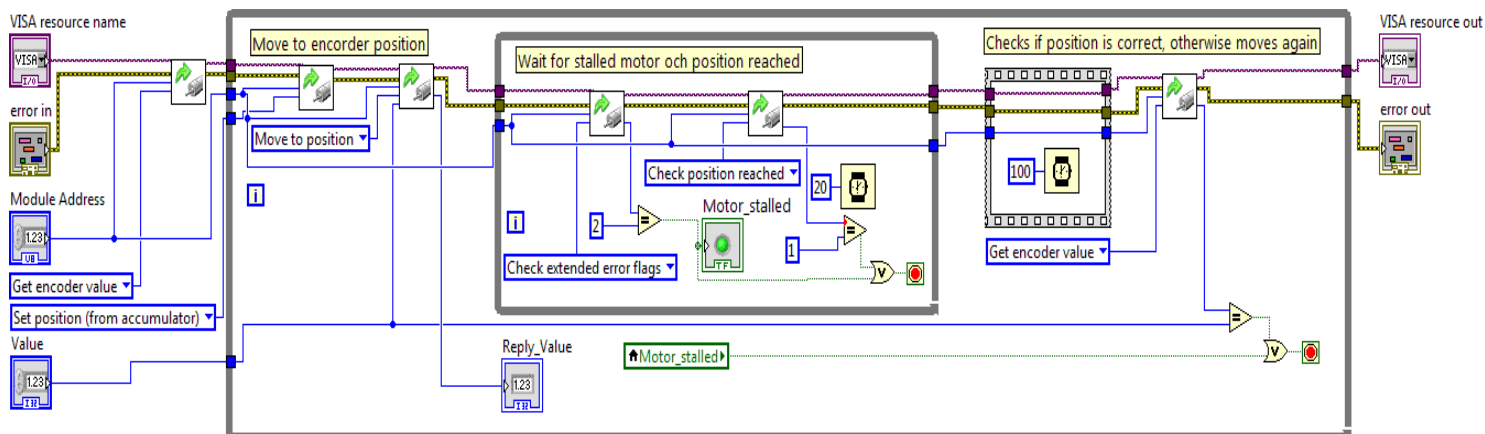


PANdrive_move_to_exact_position.vi

Moves motor to a chosen position. To make sure position is reached, it checks position when the motor has finished moving, and adjusts position if necessary.

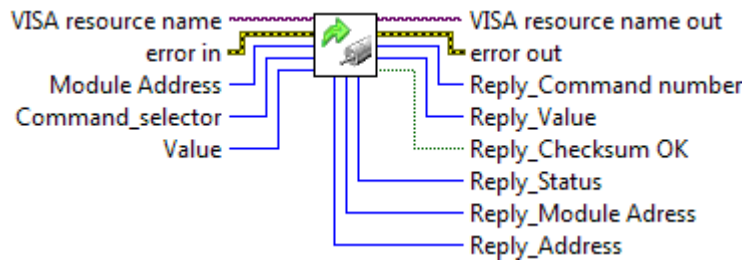


Front panel of the VISA resource name interface. It includes fields for VISA resource name, error in status and code, source, error out status and code, source, Module Address, VISA resource out, Reply_Value, and Motor_stalled.

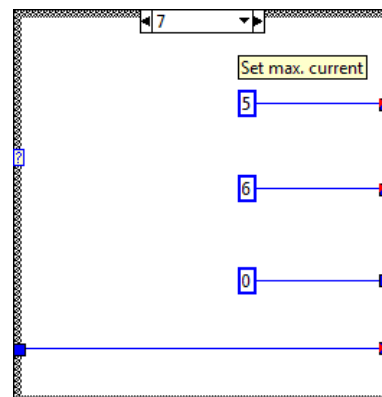


PANdrive_Send_command.vi

Sends a chosen command to a PANdrive with a chosen module address.

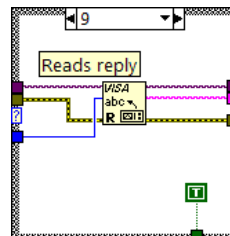
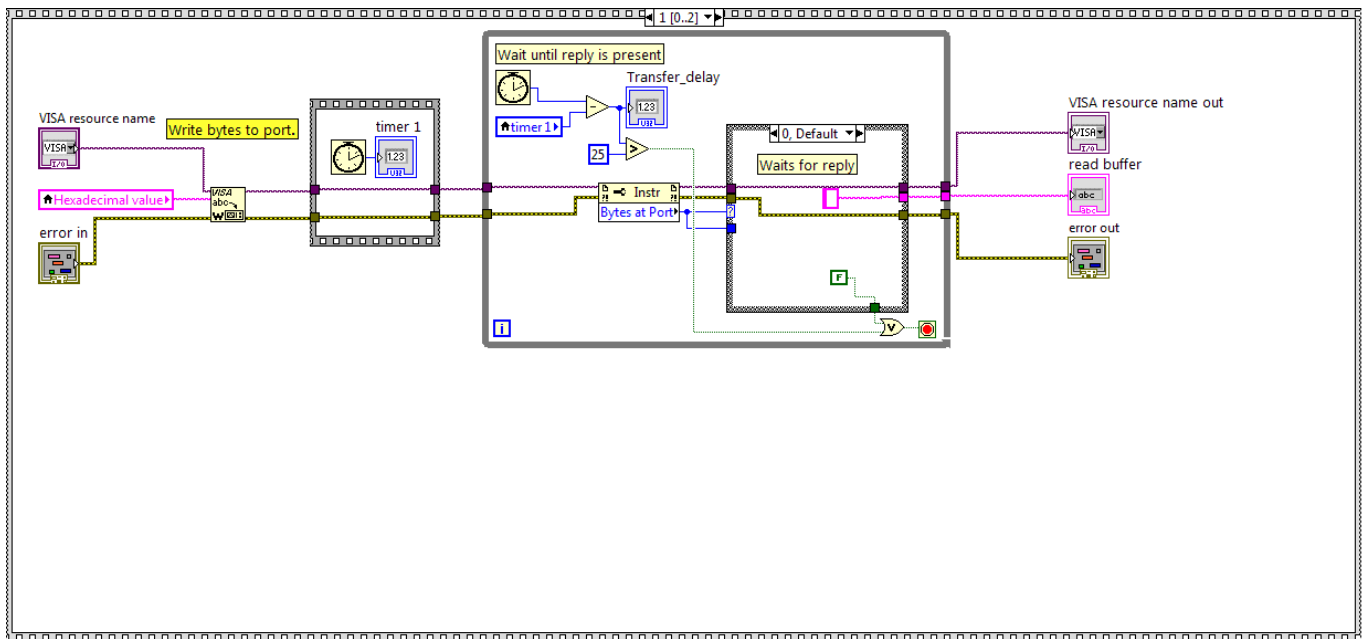
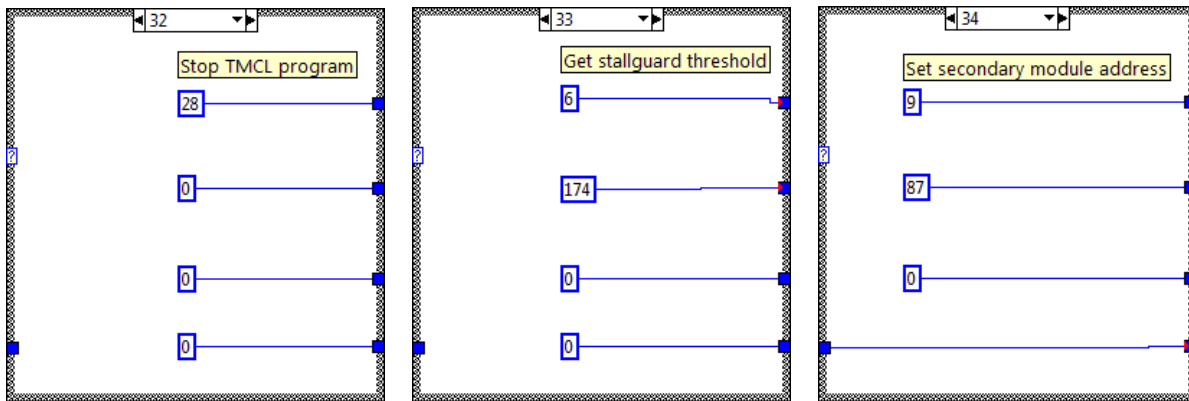


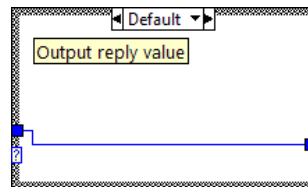
The front panel of the PANdrive_Send_command.vi VI is divided into two main sections. The left section contains input controls: a dropdown menu for 'VISA resource name' with '1' selected, a numeric spinner for 'Module Address' with '1' and a 'Value' label, a numeric spinner for 'Value' with '50', and a dropdown menu for 'Command_selector' with 'Stop' selected. The right section contains output displays: an 'error out' status indicator with a green checkmark and a 'code' display showing 'd0', a 'source' text area, a 'Transfer_delay' numeric spinner with '0', a 'Reply_Checksum OK' indicator with a green light, a 'Reply_Address' numeric spinner with '0', a 'Reply_Module Address' numeric spinner with '0', a 'Reply_Status' numeric spinner with '0', a 'Reply_Command number' numeric spinner with '0', and a 'Reply_Value' numeric spinner with '0'.





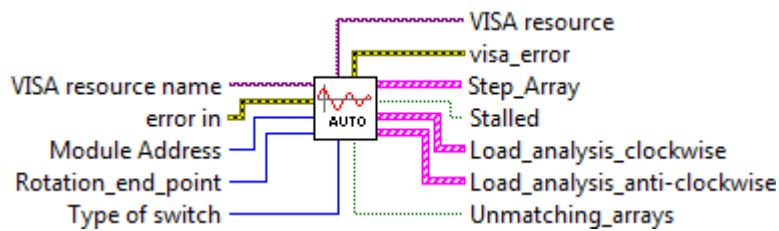




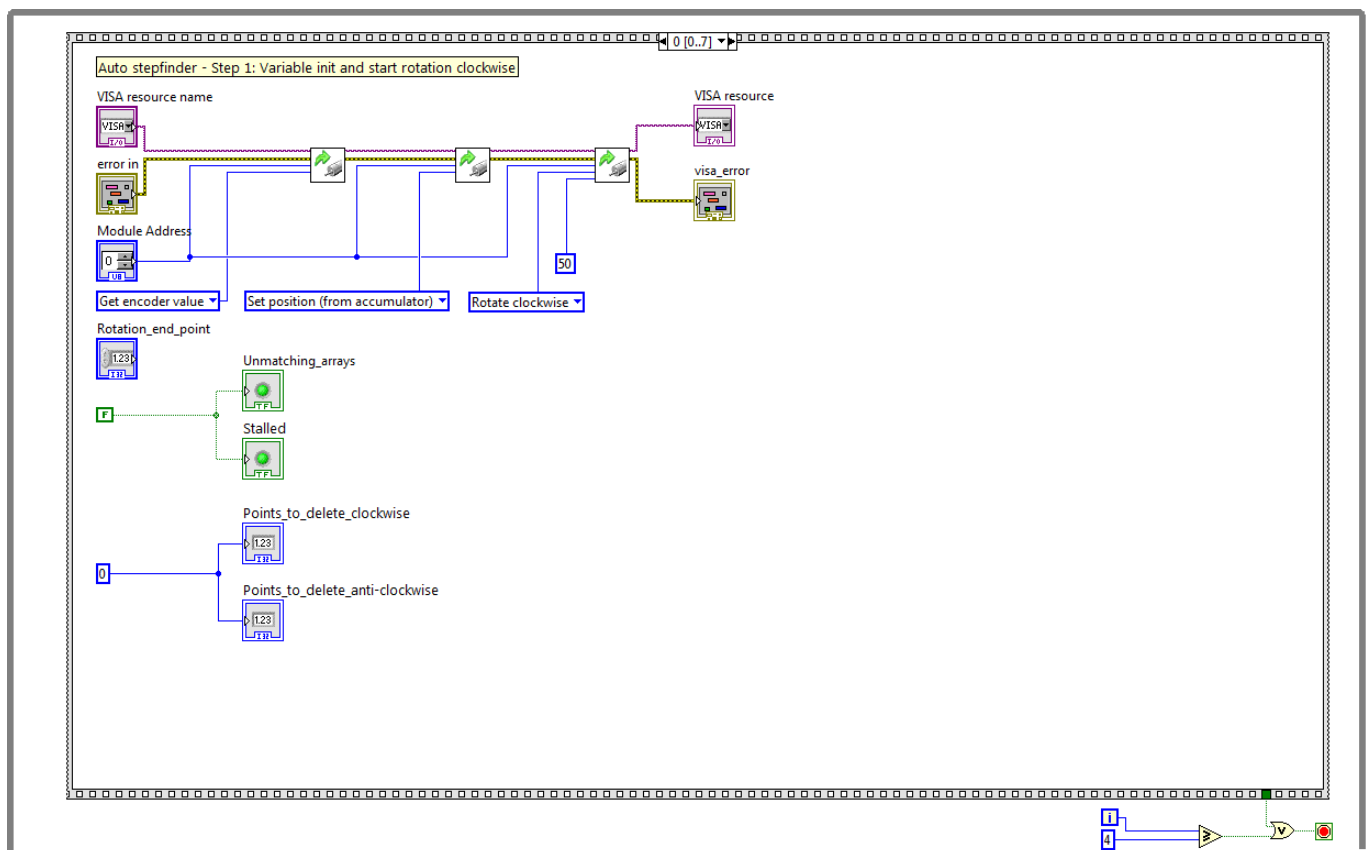


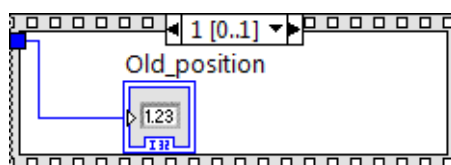
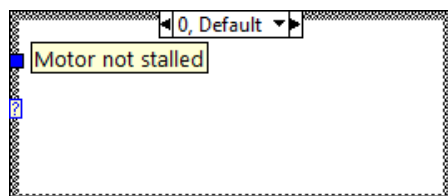
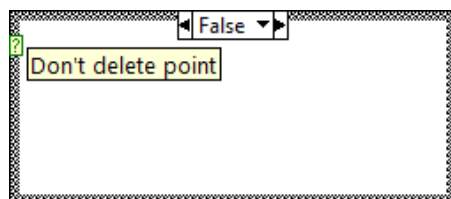
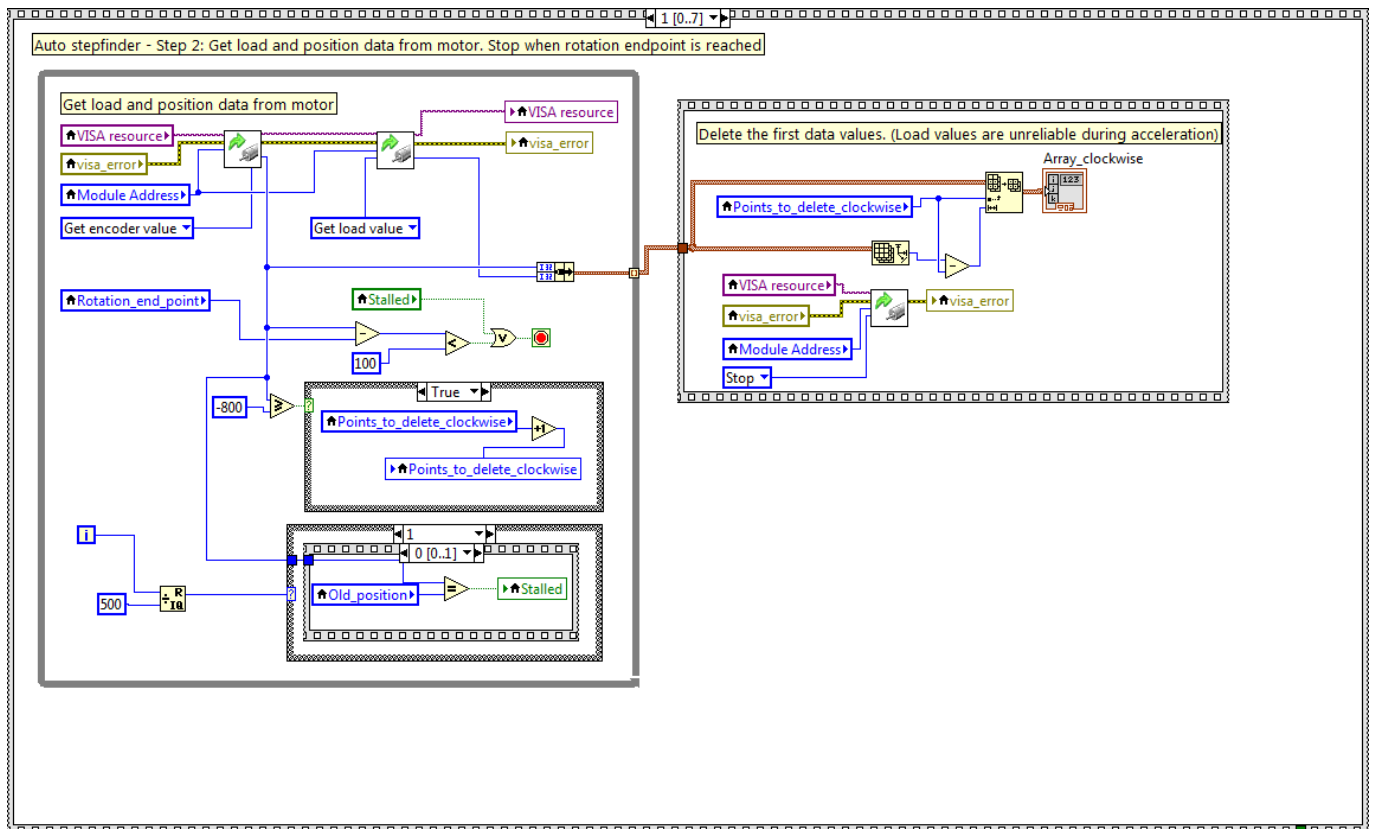
PANdrive_stepfinder.vi

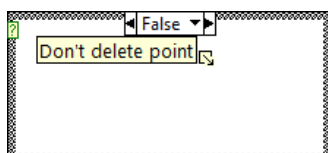
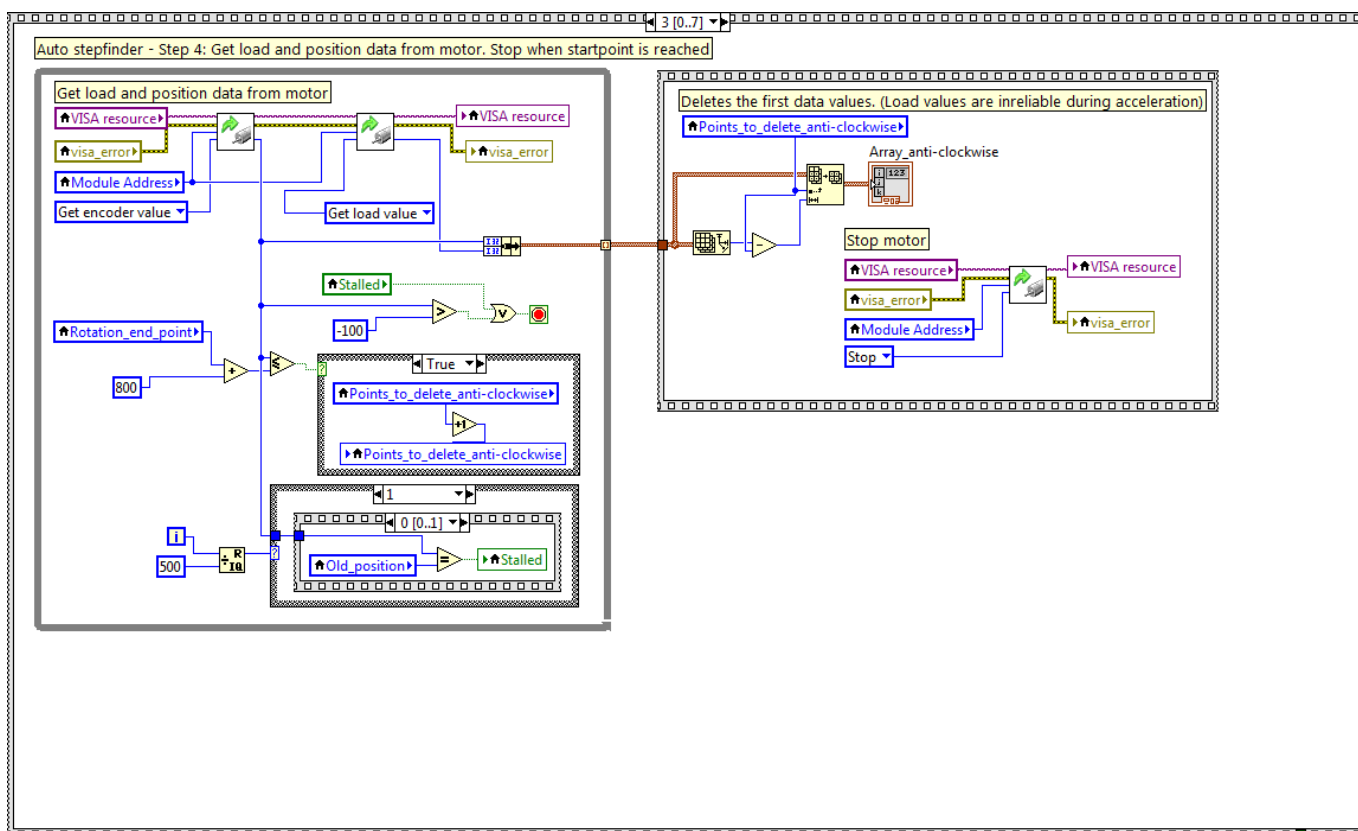
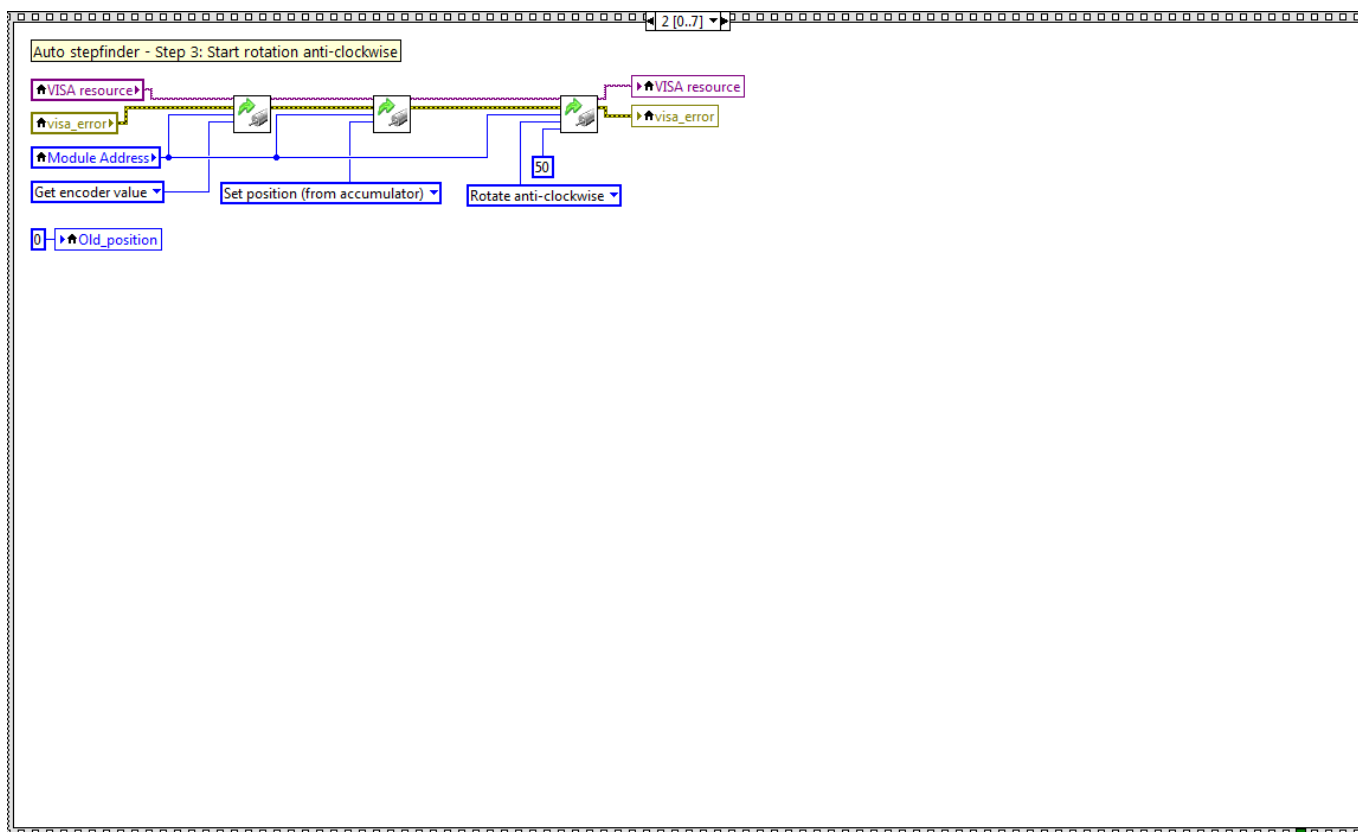
Rotates the switch through its range and collects load- and position data to pinpoint its steps.

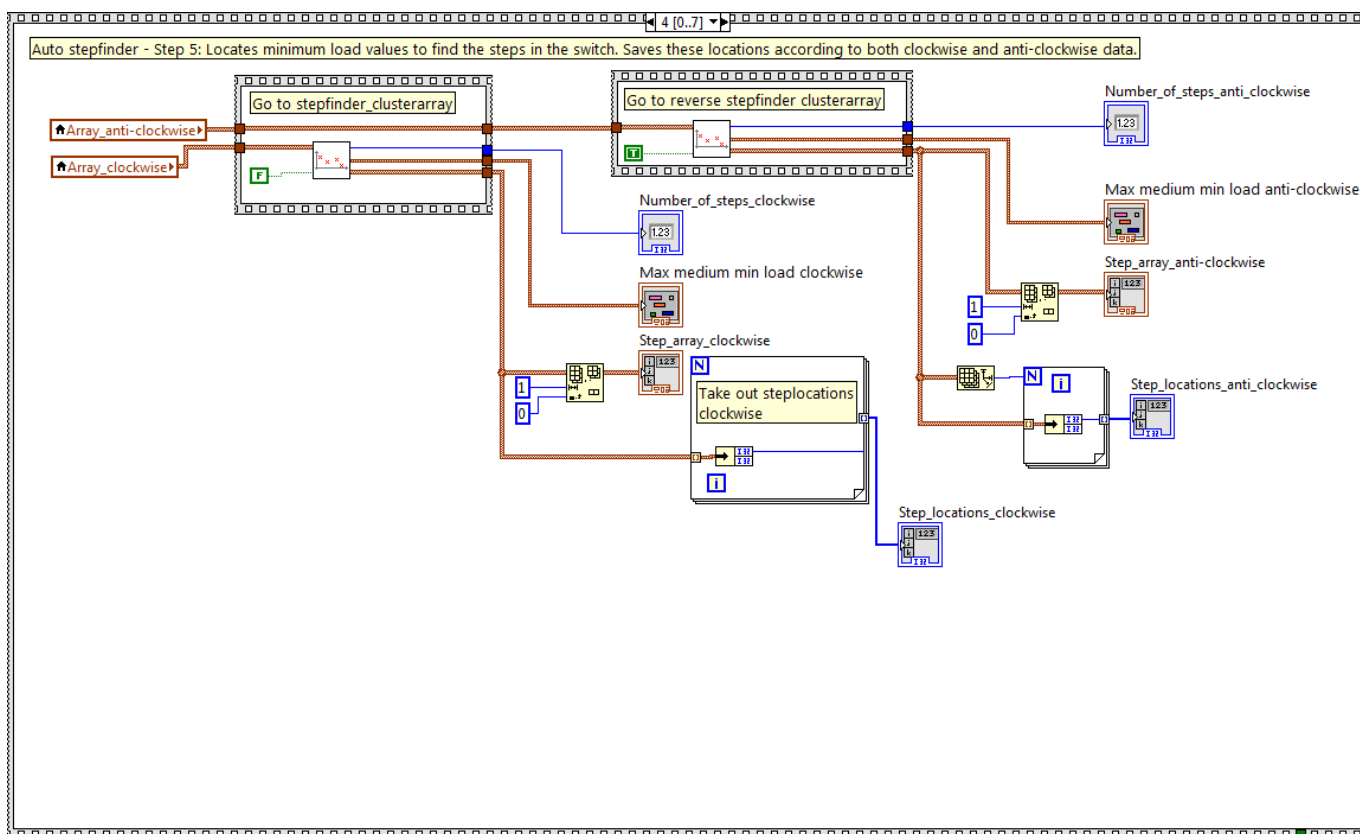
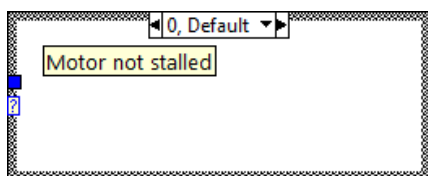


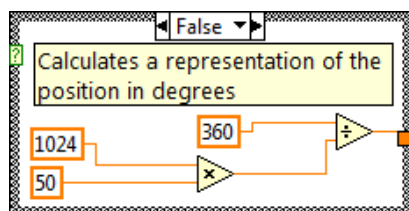
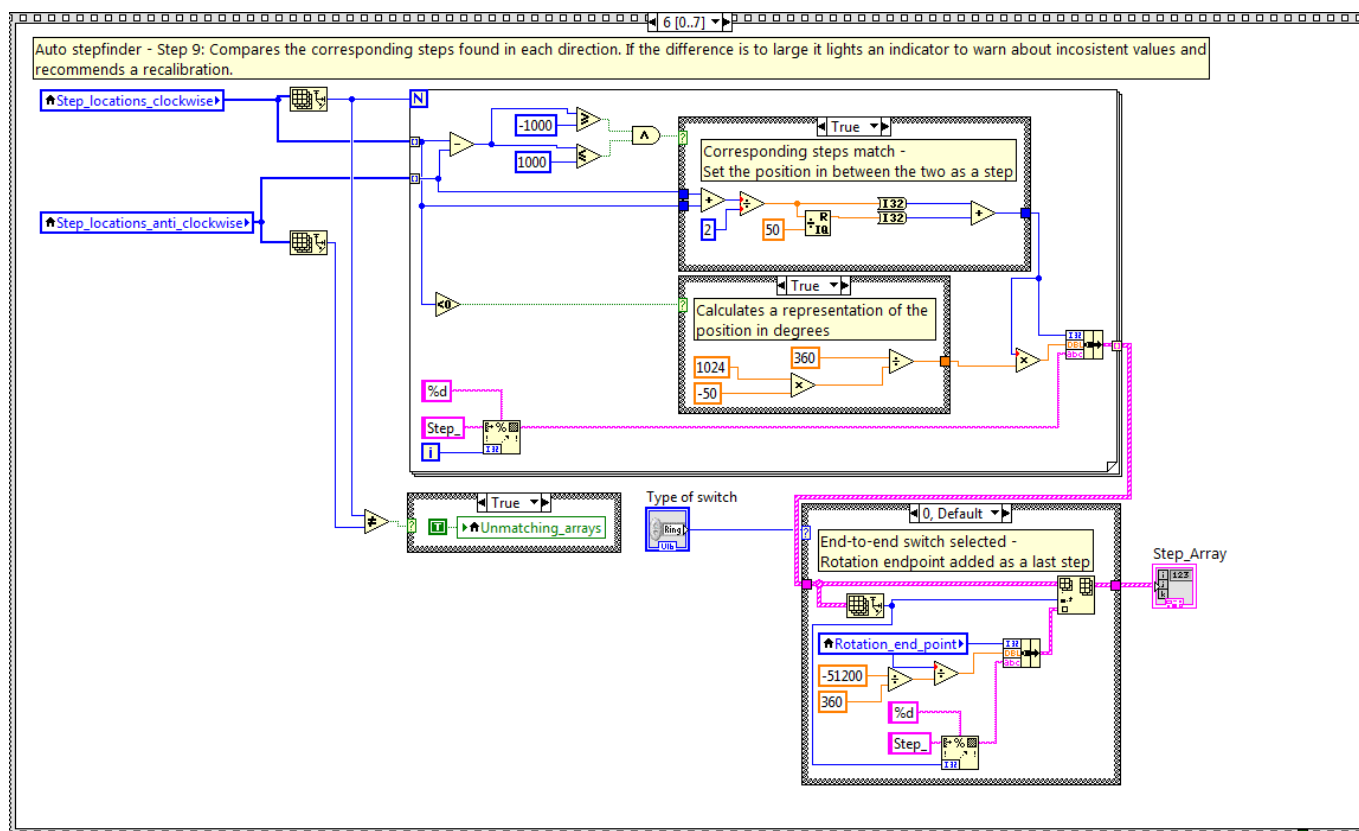
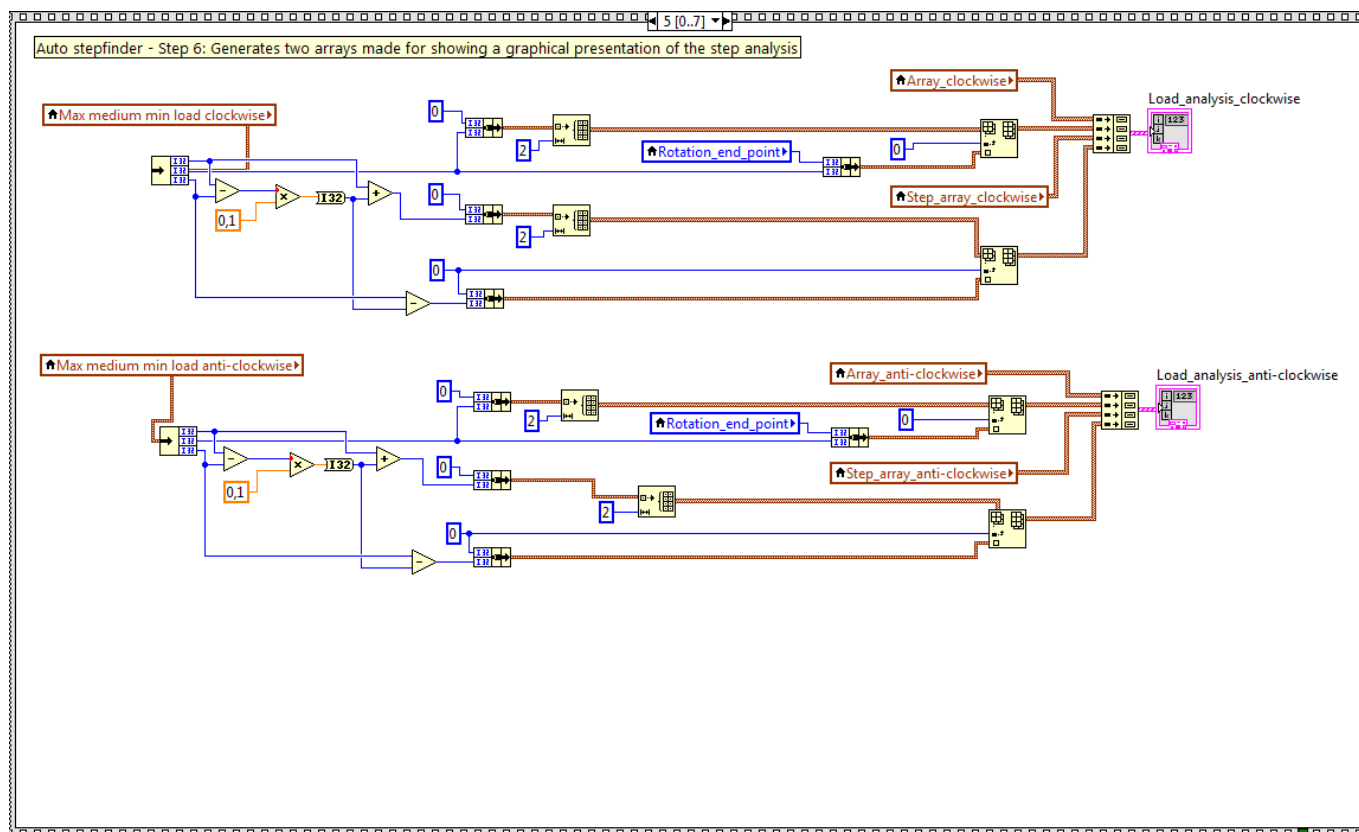
Looking for steps...

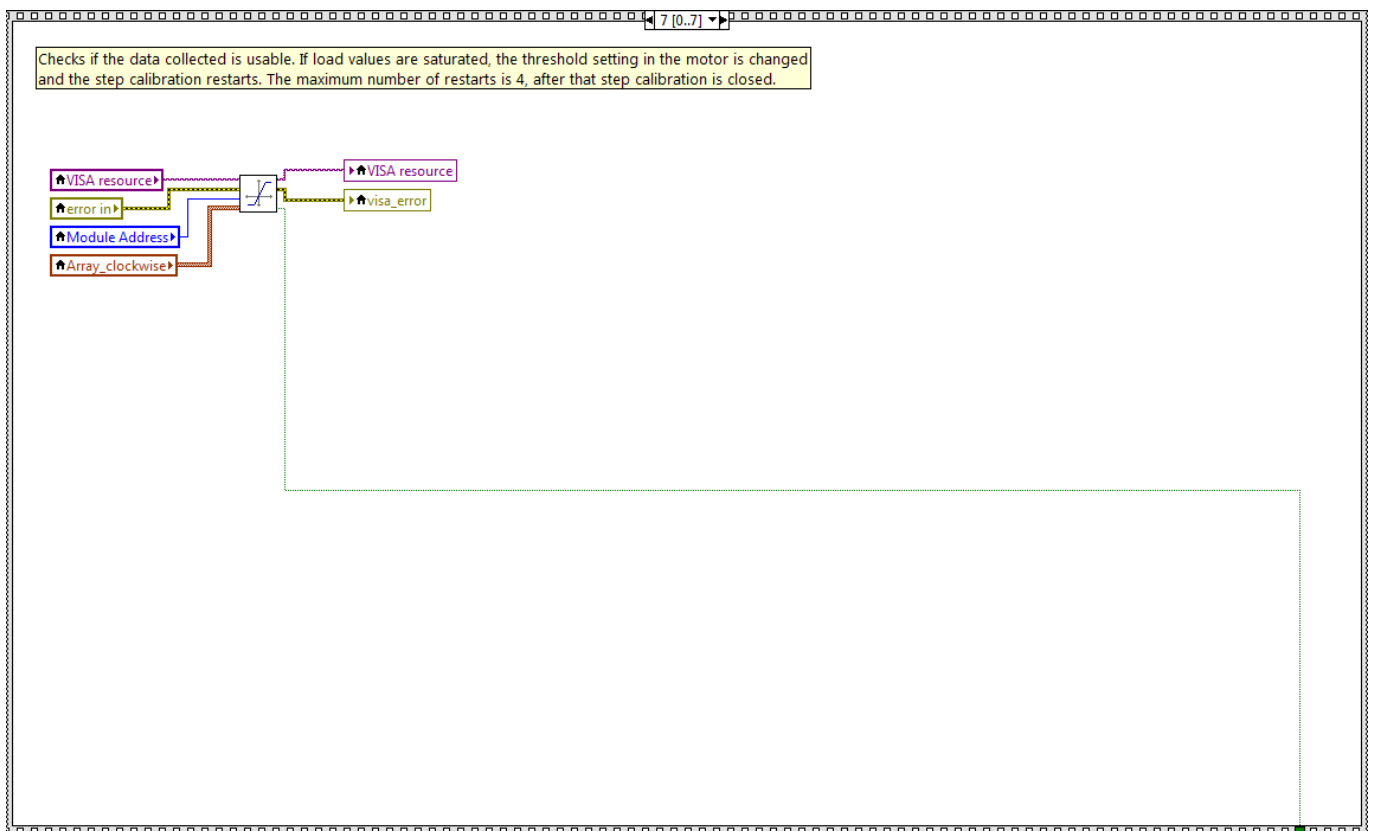






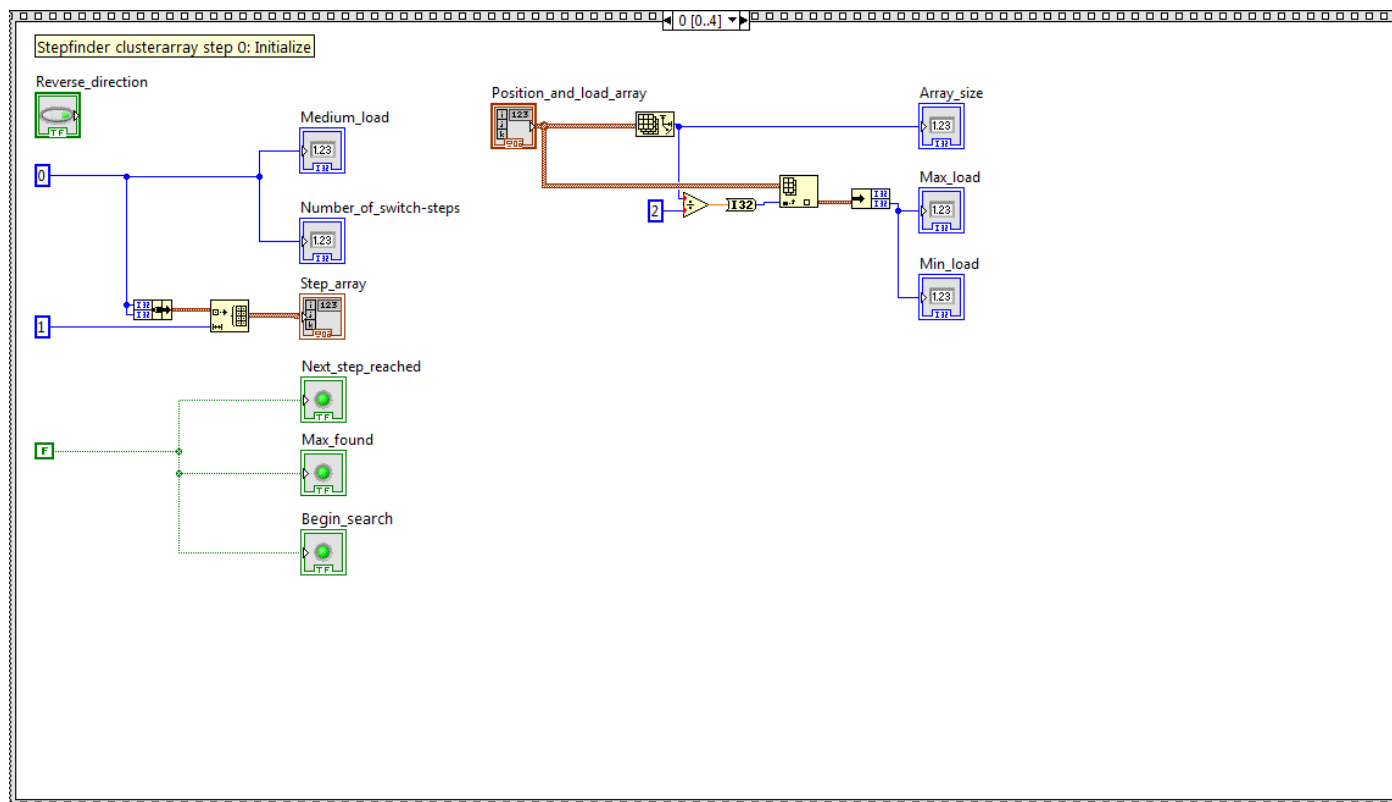
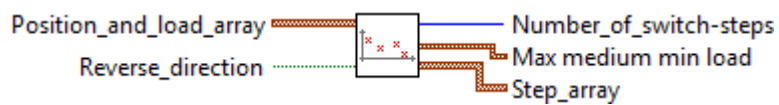


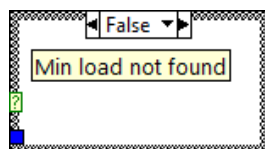
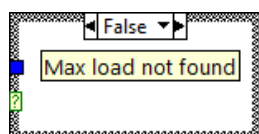
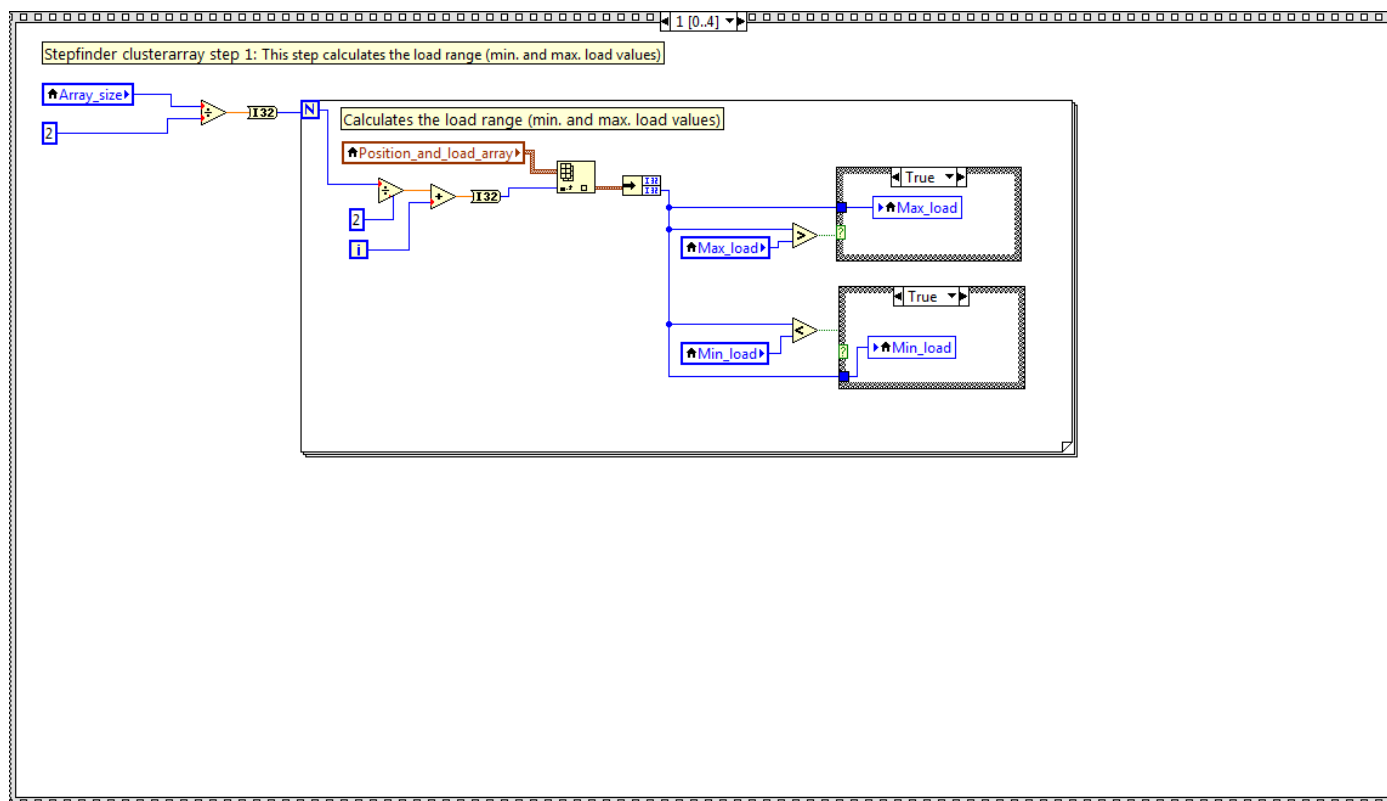


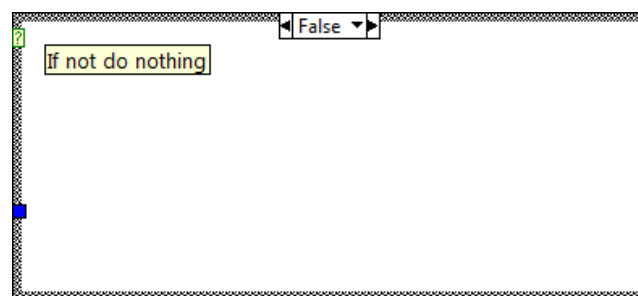
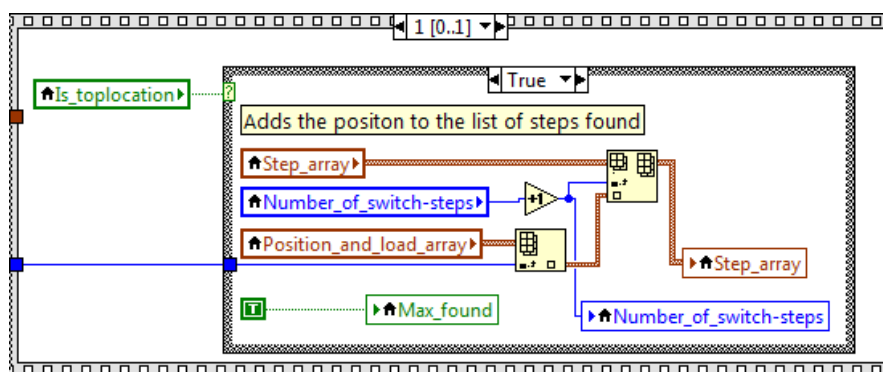
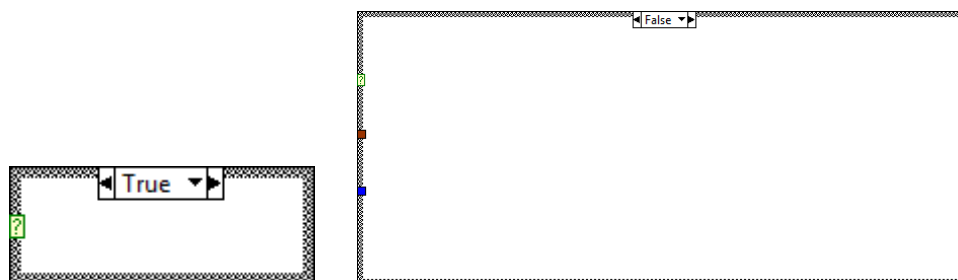
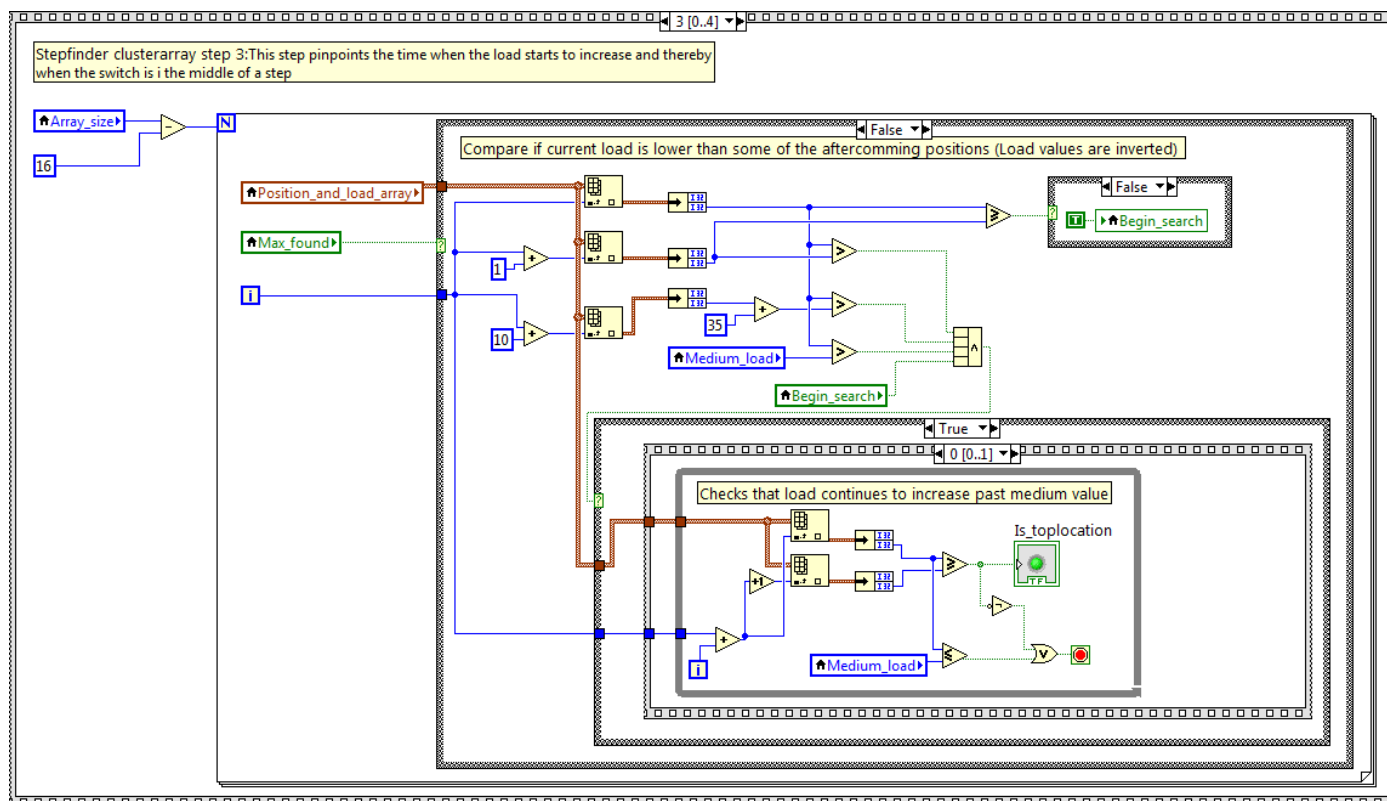


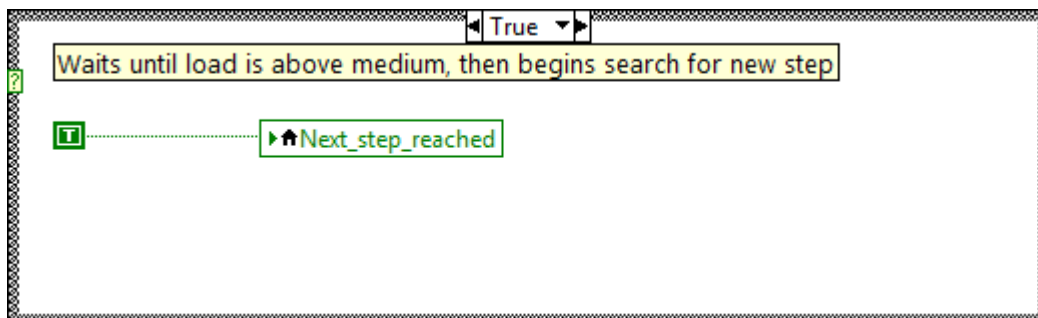
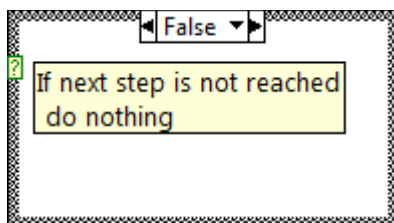
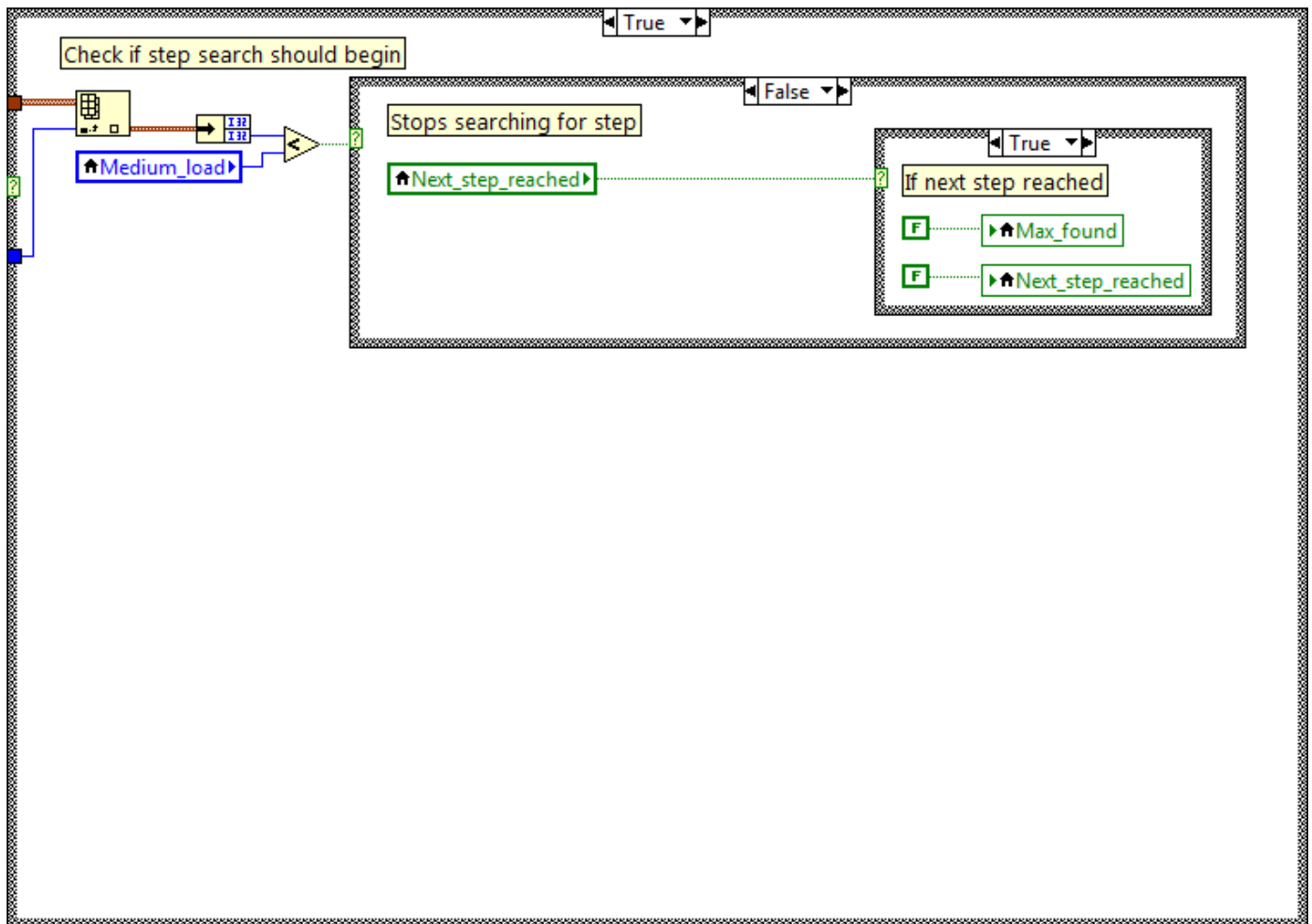
PANdrive_Stepfinder_analyze_data.vi

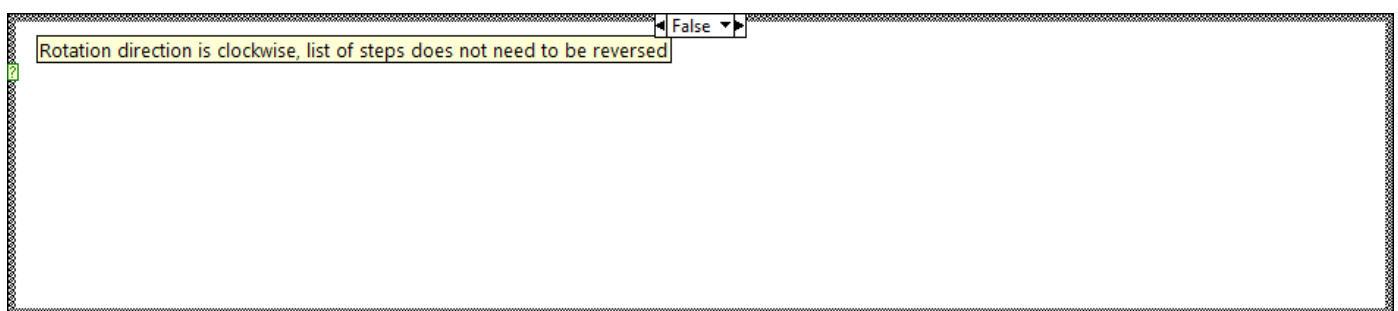
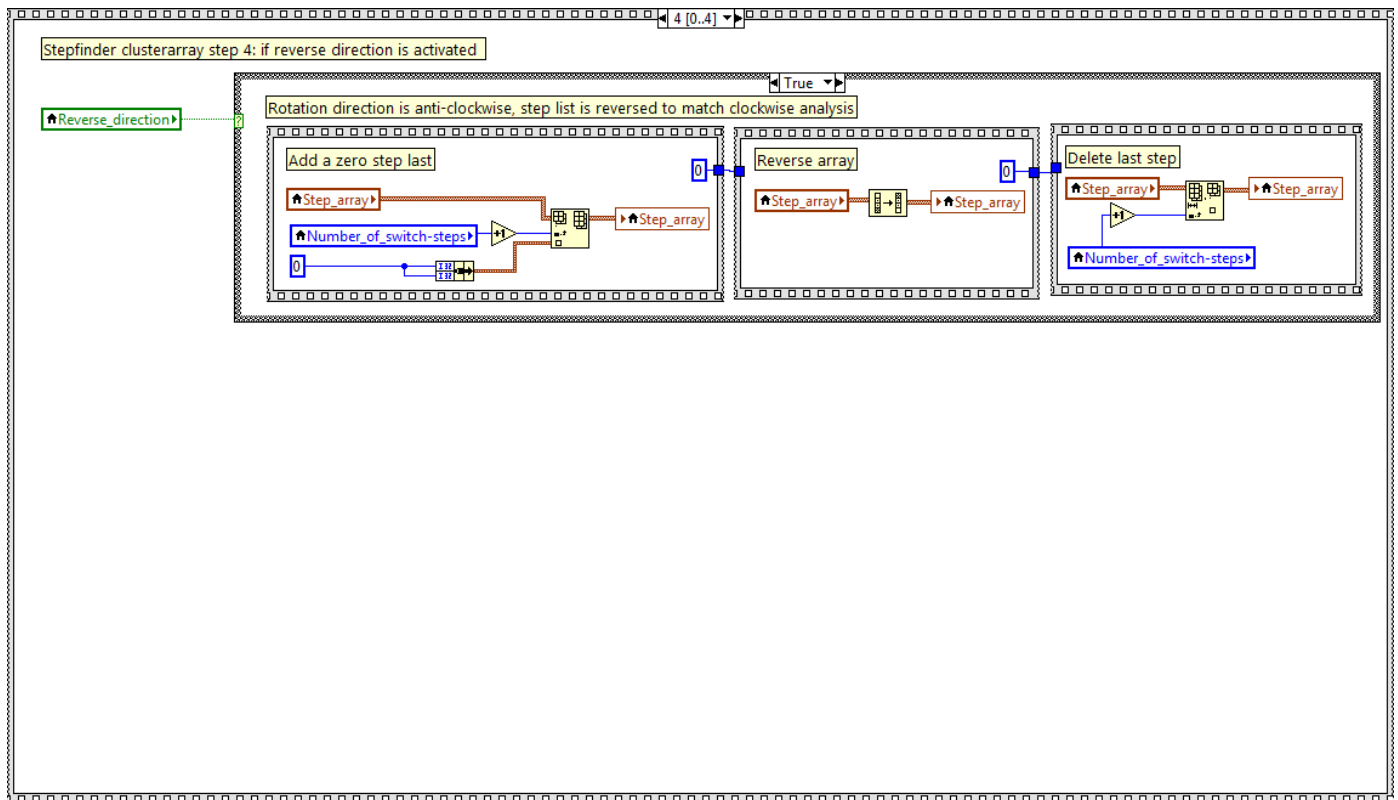
Analyzes load data to find the positions where the least torque is needed and thereby find the steps in the rotary switch.





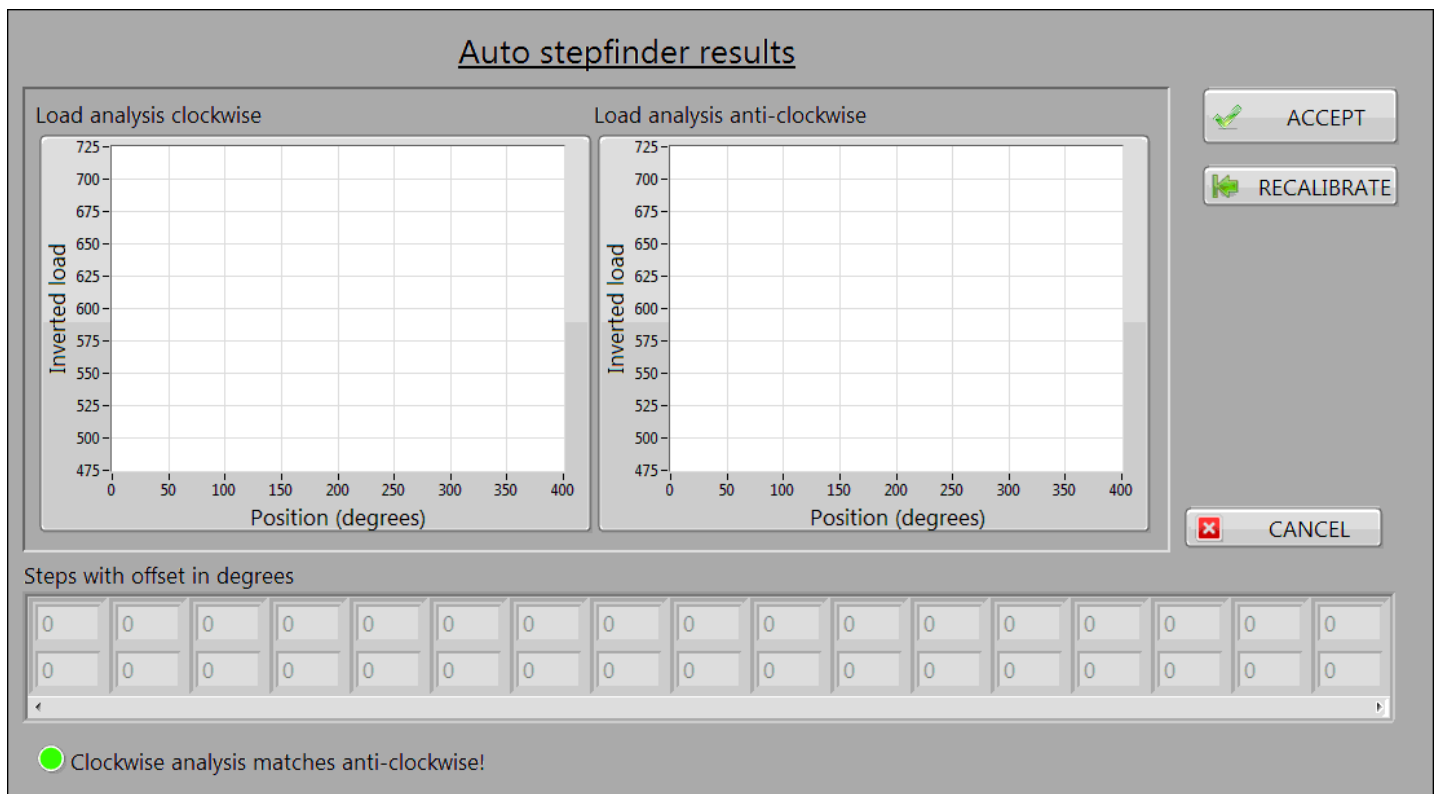
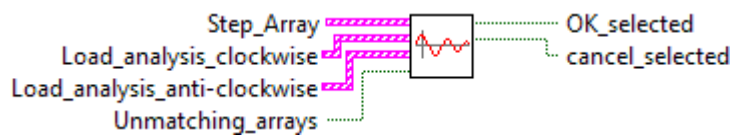


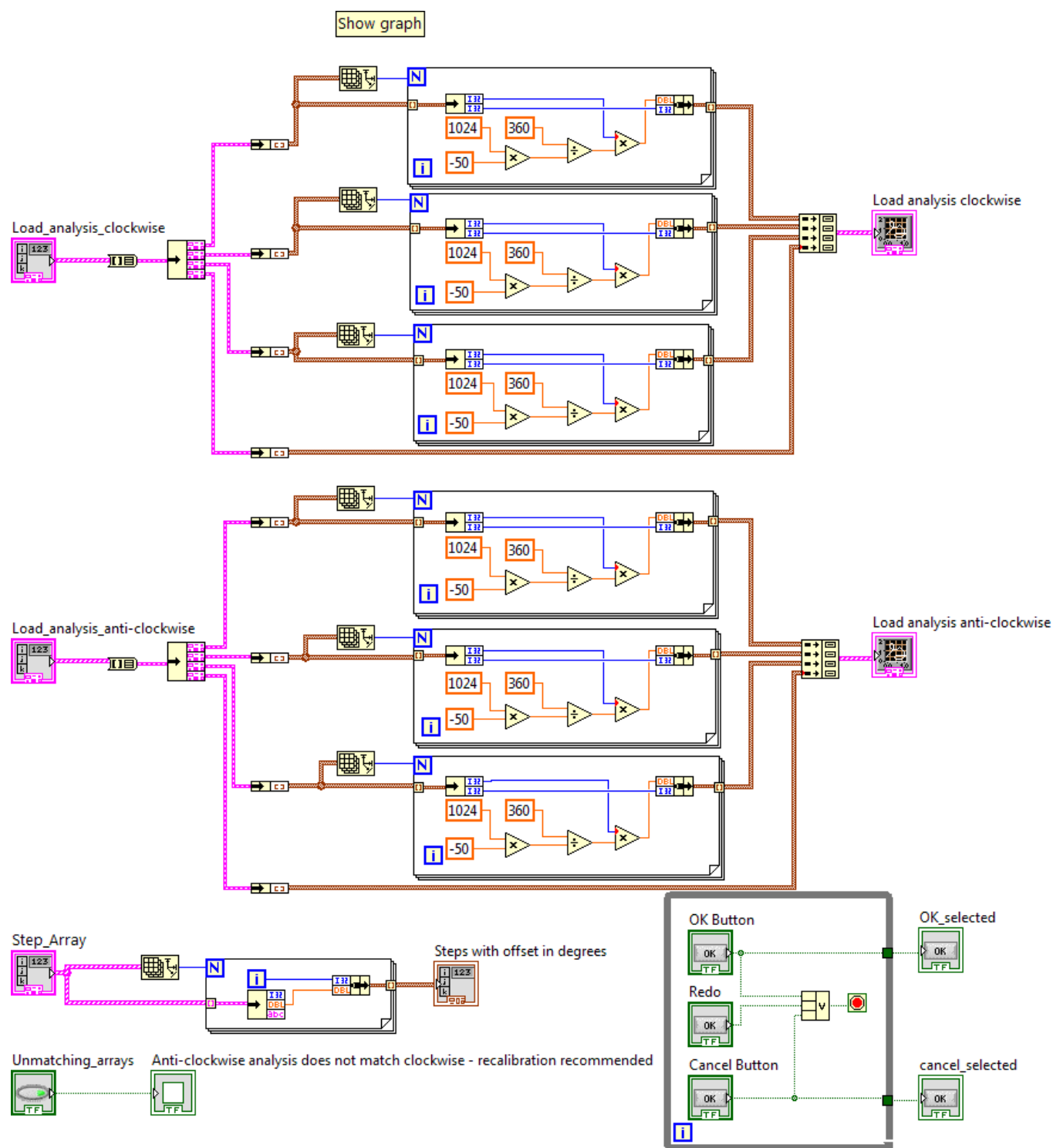




Present_stepfinder.vi

Present the data collected by the stepfinder, lets the user decide if step calibration is OK.

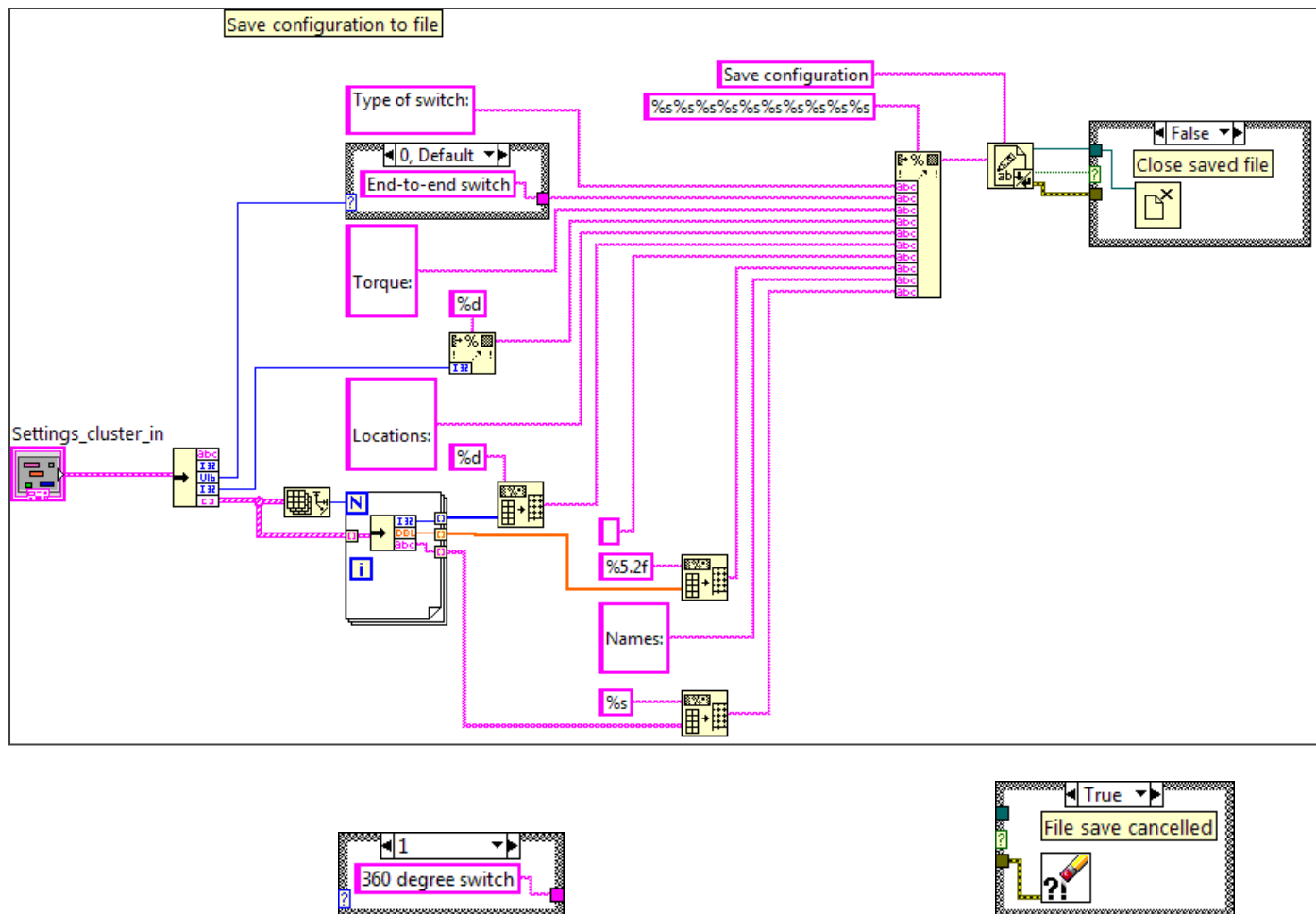




Save_config_file.vi

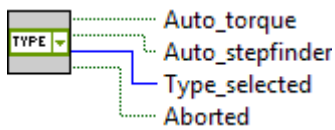
Saves a rotary switch calibration into a file.

Settings_cluster_in



Select_calibration_type.vi

Lets user input which type of switch the motor unit is attached to.



Select Calibration method

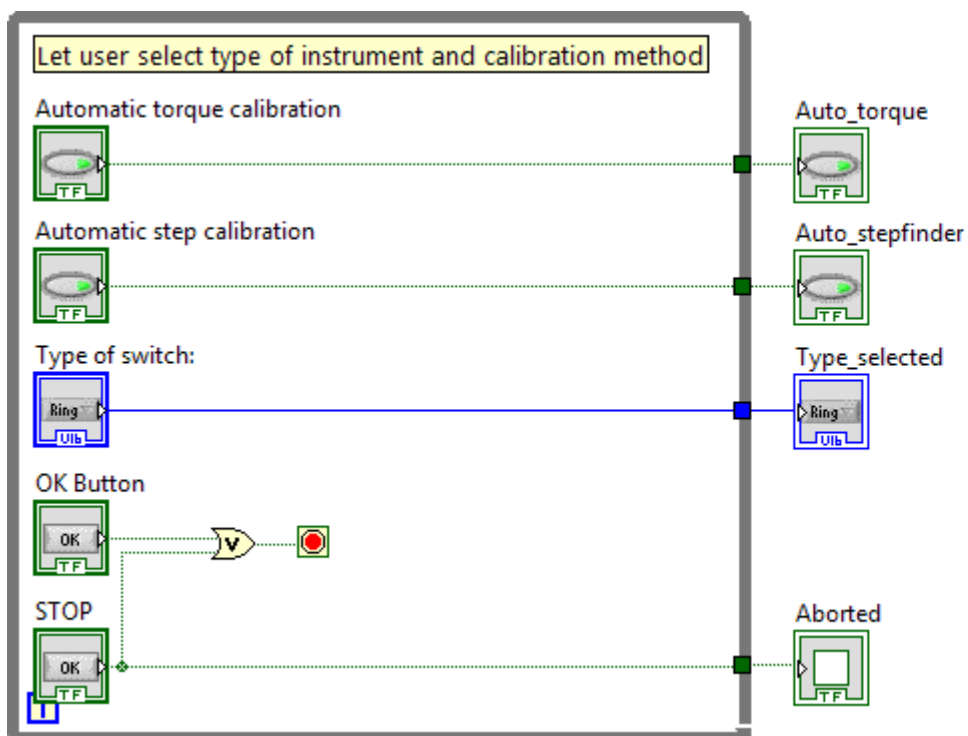
Type of switch:

End-to-end with steps

Automatic torque calibration

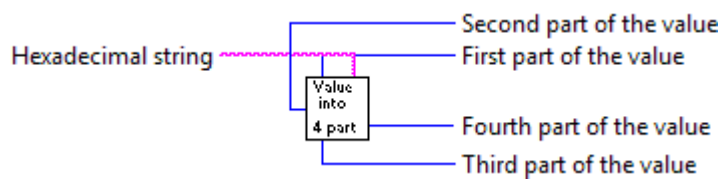
Automatic step calibration

OK CANCEL



Value in hex to 4 decimals.vi

Splits a value that is to be sent to the PANdrive into four separate bytes.



Cut hexadecimal string representing instruction value into four substrings (8 byte hexadecimal strings).

Hexadecimal string

