

# Utveckling av testprogram för CANopen inklinometer och gyroskop, AMU-GEO-360

Simon Nylund

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för el- och automationsteknik

Vasa 2019



## EXAMENSARBETE

Författare: Simon Nylund  
Utbildning och ort: El- och automationsteknik, Vasa  
Inriktningsalternativ: Automation  
Handledare: Roger Mäntylä

Titel:  
Utveckling av testprogram för CANopen inklinometer och gyroskop, AMU-GEO-360

---

Datum: 20.12.2019

Sidantal: 25

Bilagor: 1

---

### Abstrakt

Detta examensarbete har utförts på uppdrag av TK Engineering Oy under våren 2019. TK Engineering, tillsammans med deras partnerföretag, utvecklar mjukvara, hårdvara och erbjuder konsulttjänster för CAN och CANopen lösningar inom marin, fordon, järnväg, luftfart, medicin, militär, maskin etc.

Uppgiften var att planera, designa och utveckla ett testprogram för inklinometer och gyroskop, AMU-GEO-360 som kontrollerar om sensorn har blivit korrekt kalibrerad av tillverkaren och håller sig inom mätområdet.

Testprogrammet skrevs i C++ och utvecklades i Visual Studio och ett skript för att konvertera resultatfilen till kalkylblad skrevs i Python.

Resultatet blev ett lättanvänt testprogram som ger användaren noggranna mätdata från AMU-GEO-360 och ger godkänt eller underkänt resultat för sensorn.

---

Språk: Svenska

Nyckelord: TKE, testutrustning, CAN, CANopen, inklinometer, gyroskop, AMU-GEO-360

---

## OPINNÄYTETYÖ

Tekijä: Simon Nylund  
Koulutus ja paikkakunta: Sähkö- ja automaatiotekniikka, Vaasa  
Suuntautumisvaihtoehto: Automaatiotekniikka  
Ohjaaja: Roger Mäntylä

Nimike:  
CANopen kallistusmittarin ja gyroskoopin, AMU-GEO-360, testiohjelman kehittäminen

---

Päivämäärä: 20.12.2019

Sivumäärä: 25

Liitteet: 1

---

### Tiivistelmä

Tämä opinnäytetyö on tehty toimeksiantona TK Engineeringille keväällä 2019. TK Engineering kehittää kumppaniyritystensä kanssa ohjelmistoja, laitteistoja ja tarjoaa konsultointipalveluita CAN- ja CANopen-ratkaisuille meri-, ajoneuvo-, rautatie-, ilmailu-, lääketiede-, sotilas-, koneteollisuudessa jne.

Tehtävänä oli suunnitella ja kehittää testiohjelma kallistusmittarille ja gyroskoopille, AMU-GEO-360, joka tarkistaa, onko valmistaja kalibroinut anturin oikein ja pysyykö se mitta-alueella.

Testiohjelma kirjoitettiin C++:lla ja kehitettiin Visual Studioissa, ja ohjelma tulostiedoston muuntamiseksi laskentataulukoksi kirjoitettiin Pythonissa.

Tuloksena on helppokäyttöinen testiohjelma, joka antaa käyttäjälle tarkat mittaustiedot AMU-GEO-360:sta ja antaa käyttäjälle anturin hyväksytyt tai epäonnistuneet tulokset.

---

Kieli: ruotsi

Avainsanat: TKE, testausohjelma, CAN, CANopen, kallistusmittari, gyroskooppi, AMU-GEO-360

---

## BACHELOR'S THESIS

Author: Simon Nylund  
Degree Programme: Electrical Engineering, Vaasa  
Specialization: Automation  
Supervisor: Roger Mäntylä

Title: Development of Testing Program for Inclinometer and Gyroscope, AMU-GEO-360

---

Date: December 20, 2019  
Appendices: 1

Number of pages: 25

---

### Abstract

This thesis was done on the behalf of TK Engineering Oy during spring 2019. TK Engineering, together with their partner companies, develop software and hardware offering consulting services for CAN and CANopen solutions within marine, automotive, railway, aviation, medicine, military, machining etc.

The goal was to plan, design and develop a testing program for the inclinometer and gyroscope, AMU-GEO-360 to check that the sensor has been correctly calibrated by the manufacturer and stays within the measuring range.

The testing program is written in C++ and developed in Visual Studio and a script to convert the result file to a spreadsheet was written in Python.

The result was an easy to use testing program that gives the user accurate measuring data from AMU-GEO-360 and gives a passed or failed result for the sensor.

---

Language: Swedish

Key words: TKE, testing equipment, CAN, CANopen, inclinometer, gyroscope, AMU-GEO-360

---

## Innehållsförteckning

1	Inledning.....	1
1.1	Uppdragsgivare.....	1
1.2	Uppgiftsbeskrivning.....	1
1.2.1	Bakgrund.....	1
1.2.2	Syfte .....	2
1.2.3	Mål.....	2
2	Teori.....	2
2.1	AMU-GEO-360.....	2
2.2	CAN .....	4
2.2.1	Uppbyggnad av CAN .....	4
2.2.2	DS-301.....	8
2.3	Testprocess.....	11
2.3.1	Exempeltest .....	13
3	Utvecklingsmiljö .....	14
3.1	Microsoft Visual Studio .....	14
3.2	Notepad++.....	14
3.3	Programspråk.....	14
3.3.1	C++.....	14
3.3.2	Python.....	14
4	Utförande.....	15
4.1	Krav.....	15
4.2	Planering.....	15
5	Resultat .....	20
6	Diskussion.....	24
7	Källförteckning.....	24

## Figurförteckning

Figur 1, Stötkompensering. [3].....	3
Figur 2, Vibrationsstabilisering [3].....	3
Figur 3, OSI 7-skikt referensmodell. [4].....	4
Figur 4, CAN-nätverk topologi. [8] .....	5
Figur 5, CAN bas format för data frames. [4] .....	6
Figur 6, CAN- och CANopen-protokoll [4].....	7
Figur 7, Object Dictionary-post i Electronic Data Sheet-fil. [4].....	9
Figur 8, Network Management-tillstånd för slave-noder. [4] .....	10
Figur 9, Testjigg .....	11
Figur 10, AMU-GEO i tre olika lägen för test. ....	12
Figur 11, Testjigg roteras medsols.....	12

Figur 12, CANtrace fönster.....	13
Figur 13, Flödesschema del 1 (initialisering av AMU-GEO).....	16
Figur 14, Flödesschema del 2 (kontroll av angle360 (AMU-GEO:s vinkel)). .....	16
Figur 15, Flödesschema del 3. (kontroll av vinkelhastighet).....	17
Figur 16, Flödesschema del 4 (sparande av startvärde innan rotering).....	17
Figur 17, Flödesschema del 5.....	18
Figur 18, Flödesschema del 6.....	18
Figur 19, Flödesschema del 7 (Utskrivning samt sparande av testresultat).....	19
Figur 20, Flödesschema för konverteringskript. ....	20
Figur 21, Startmeny. ....	21
Figur 22, angle360 indikator.....	21
Figur 23, Mellanrapport med resultat från deltest 1. ....	22
Figur 24, Misslyckat testförsök. ....	22
Figur 25, Resultat från ett komplett test.....	22
Figur 26, Skriptet skapar kalkylark. ....	23
Figur 27, Kalkylark med resultat från 4 sensorer. ....	23
Figur 28, Kalkylark med resultat från 4 sensorer (forts.).....	23

## Tabellförteckning

Tabell 1, CANopen Frameworks. [4] .....	7
Tabell 2, Device Profiles. [4] .....	8
Tabell 3, Indexering av Object Dictionary. [4] .....	9
Tabell 4, Kommunikationstyper aktiva vid olika tillstånd. [4].....	11
Tabell 5, Start- och slutvärden från rotationstest. ....	13

## Förkortningar

TKE	TK Engineering Oy
CAN	Control Area Network
CiA	CAN in Automation
ISO	International Organization for Standardization
OSI	Open Systems Interconnection
LLC	Logical Link Control
MAC	Media Access Control
COB	Connection Object
CRC	Cyclic Redundancy Checksum
DS	Draft Specification
DSP	Draft Standard Proposal

OD	Object Dictionary
EDS	Electronic Data Sheet
SDO	Service Data Object
PDO	Process Data Object
NMT	Network Management
RTR	Remote Transmission Request

# 1 Inledning

Detta examensarbete handlar om skapandet av ett testprogram åt TK Engineering Oy för att kontrollera gyroskop och inklinometer, AMU-GEO-360. Programmet ska kontrollera sensorernas noggrannhet och säkerställa att sensorn har blivit korrekt kalibrerad av tillverkaren.

## 1.1 Uppdragsgivare

Uppdragsgivare till examensarbetet var TK Engineering Oy. TKE är beläget i centrum av Vasa och består av en lokal. Företaget grundades år 2003 av Timo Kesti, har idag åtta anställda utöver Timo och har en årlig omsättning på omkring en miljon euro. [1] [2]

TKE är ett ingenjörföretag som fokuserar på utbildning, teknik, elektronik och mjukvaruutveckling med erfarenhet av många olika CAN-relaterade projekt och arbetar med kunder från olika industrier runt om i världen, såsom marin, materialhantering, fordon, järnväg, maskiner, luftfart, medicin, militär, forskningsinstitutioner etc. Tillsammans med sina partnerföretag erbjuder TKE aktuella och moderna CAN-lösningar. Tjänsterna omfattar utveckling av ny hårdvara och mjukvara efter kundernas behov, med sina tjänster och produkter hjälper TKE kunden från design och utveckling till en färdig produkt. TKE säljer även färdiga sensorer, konfigurerings- och analyseringsprogram för CAN och CANopen samt styrkort och gränssnitt av tredje parter. [2]

## 1.2 Uppgiftsbeskrivning

Uppgiften var att planera och utveckla ett testprogram som försnabbar och strömlinjeformar kontrollprocessen av AMU-GEO-360:s kalibrering.

### 1.2.1 Bakgrund

Förut har TK Engineering varit tvungna att kontrollera AMU-GEO-360:s kalibrering genom att använda analyseringsprogram för CAN-data, ta skärmbilder av start- och slutvärden, använda stoppur för att hålla reda på testtider, rotera testjigg samt manuellt fylla i alla testresultatdata i kalkylark. Detta var en långsam och arbetsam process som krävde att man skulle sortera och namnge alla skärmbilder samt kontrollera en gång till att man fyllt i rätt resultatdata i kalkylarken. Denna process går igenom i kapitel 2.3.



### 1.2.2 Syfte

Syftet med examensarbetet var att förbättra och försnabba företagets process att kontrollera om tillverkaren har kalibrerat inklinometer och gyroskop, AMU-GEO-360, korrekt. Den dåvarande kontrollprocessen tog upp för mycket tid vilket gjorde att denna produkt var olönsam för TKE. Därför var det viktigt att försnabba och strömlinjeforma denna process så mycket som möjligt.

### 1.2.3 Mål

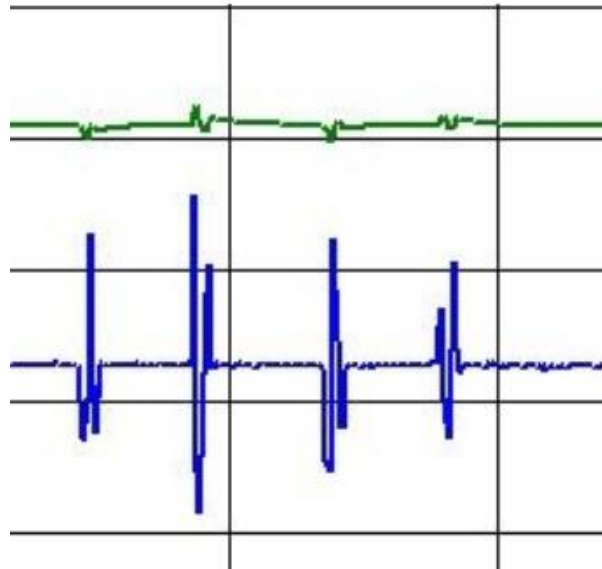
Målet med arbetet var ett lättanvänt testprogram som säger åt användaren vad som måste göras och när, ger noggranna mätdata och skriver en resultatfil. Programmet ska läsa serienummer, mjukvaruversion, vinkelhastighet och vinkel från AMU-GEO-360 och med hjälp av dessa data kontrollera om sensorn hålls inom mätgränserna. Ett skript används för att skriva om resultatfilerna till ett kalkylark som är lättare att läsa och ger ett slutgiltigt resultat för varje batch om vilka sensorer som klarat testet eller misslyckats och måste skickas tillbaka till tillverkaren.

## 2 Teori

I detta kapitel förklaras den allmänna teorin om teknikerna, hårdvaran och mjukvaran relaterat till examensarbetet. Teorin kommer behandla vad, gyroskop och inklinometer, AMU-GEO-360 är hur den fungerar samt vad databussen CAN och CANopen är och hur man använder dessa.

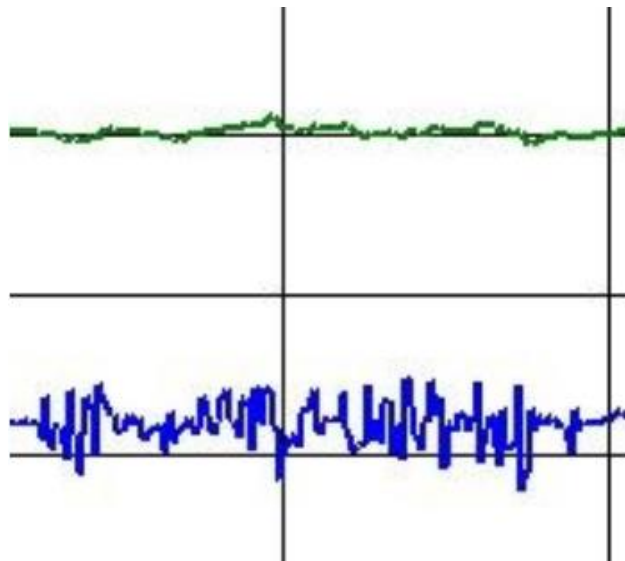
### 2.1 AMU-GEO-360

AMU-GEO-360 är en kompenserad tröghetsensor som är kapabel till att mäta ett fordon's position och hastighet i rymden. AMU-GEO används ofta på lyftkranar för precision eller grävmaskiner som säkerhetsåtgärd så att föraren inte kör skopan in i hytten. Den består av en 3-axlig accelerometer, gyroskop och kompass, för att få 3-axligt geometriskt läge och 2-axligt rotationsläge i realtid för objektet som sensorn är installerad på. Sensorn utnyttjar även mjukvara för att kompensera stötar, vibrationer, termiska chocker och centrifugalacceleration. Detta tillåter en noggrannhet på  $\pm 0,7^\circ$  under dynamiska förhållanden och  $\pm 0,3^\circ$  under statiska förhållanden i hela arbetsområdet. Grafen nedan visar hur stötar elimineras i signalen. Den gröna linjen är kompenserad och den blåa är från en traditionell accelerometerbaserad vinkelsensor.



**Figur 1, Stötkompensering. [3]**

Följande graf visar hur gyro-stabiliseringen eliminerar vibrationer. Gröna linjen är den stabiliserade signalen och blåa är från en traditionell accelerometerbaserad och filtrerad vinkelsensor.



**Figur 2, Vibrationsstabilisering [3]**

Eftersom AMU-GEO har en drifttemperatur från  $-40^{\circ}\text{C}$  till  $+80^{\circ}\text{C}$ , strömförsörjningsområde på 9 till 32V och IP67-skyddsklass så är den lämplig för mobila tillämpningar i hårda miljöer. AMU-GEO är en CANopen-certifierad inklinometer och kommunikationen följer CiA-301 och CiA-305 specifikationerna. Inklinometer-specifika gränssnittet följer CANopen-enhetsprofilen för inklinometer, CiA-410. Det fysiska kommunikationsskiktet stämmer

överens med ISO 11898–2 och inklinometerns 5-pin M12-anslutningar följer specifikationen CiA-303-1. [2] [3]

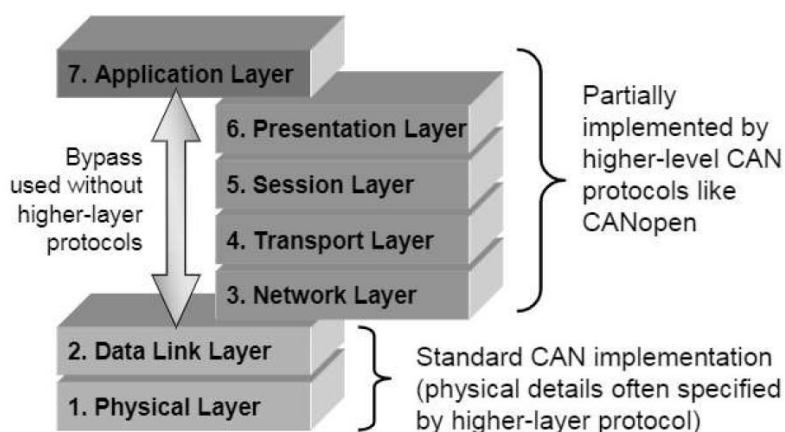
## 2.2 CAN

CAN, även kallat CAN-buss står för Controller Area Network och utvecklades under 1980-talet av Robert Bosch för bilindustrin. CAN-bussen används förutom i bilindustrin även inom tåg-, marin- och flygindustrin samt medicinsk och teknisk utrustning. Den första publicerade CAN-specifikationen heter CAN 2.0 Specification Part A. Senare släpptes en uppdaterad version, CAN 2.0 Specification Part B, som är kompatibel med den första. [4]

I början av 1992 bildades den internationella föreningen CAN in Automation (CiA) i Tyskland. Syftet med CiA är att främja utvecklingen av CAN-protokollet. CiA publicerade specifikationer för det fysiska skiktet, applikationsskiktet och enhetsprofiler. Idag är 688 företag medlemmar i CAN in Automation. [4] [5]

### 2.2.1 Uppbyggnad av CAN

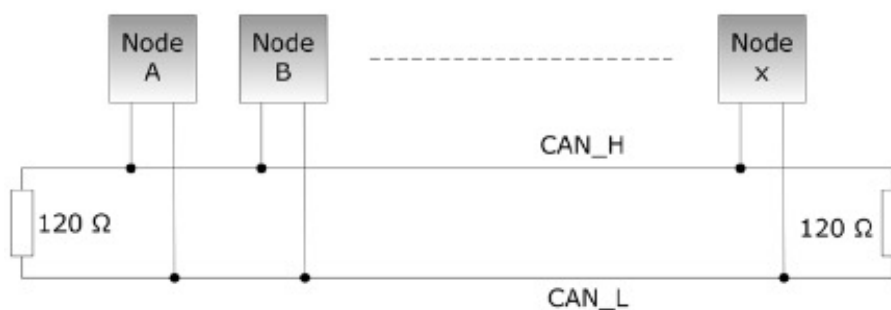
År 1993 blev CAN en ISO-standard och ISO-11898 standarden, baserad på CAN 2.0-specifikationen, är den mest använda standarden för det fysiska skiktet inom CAN-nätverk. Inom CAN-protokoll av högre nivå så som CANopen implementeras endast delar av de högre skikten från OSI-modellen (se figur nedan), eftersom protokollen har färdiga specifikationer för kommunikationen. [4] [6]



Figur 3, OSI 7-skikt referensmodell. [4]

Det första skiktet, fysiska skiktet, definierar de fysiska och elektriska specifikationerna för anordningen, så som vilken sorts kabel, kretskort, pin layout (t.ex. hur kontakterna i en stöpsel kopplas) etc. [7]

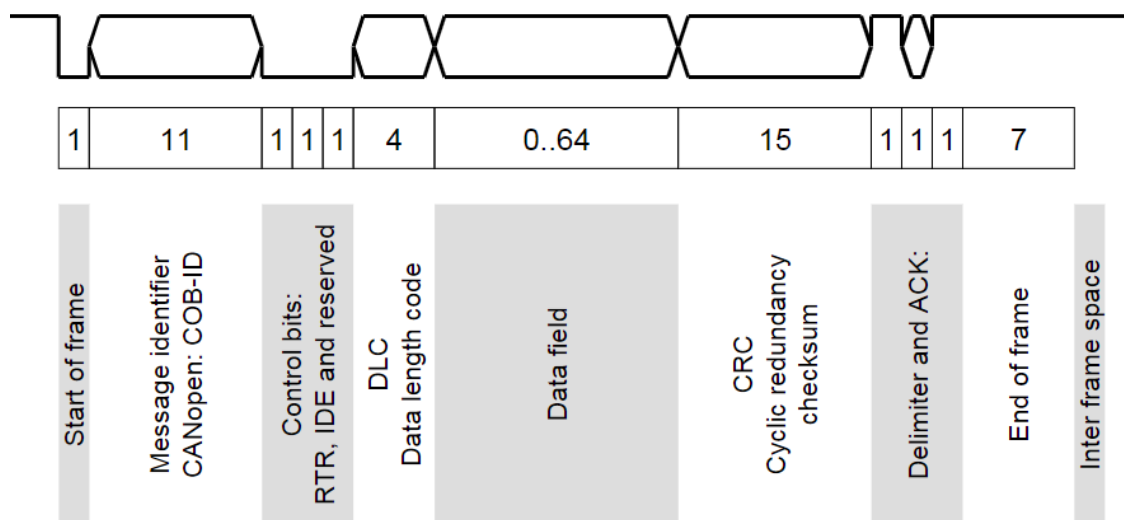
Inom CAN specificerar det bit-kodning, bit-timing samt synkronisering och definierar de fysiska medierna som används för kommunikationen i CAN-bussen. Det fysiska lagret anges i detalj i standarden ISO-11898. CAN-protokollet rekommenderar att bägge ändarna av bussen ihopkopplas med resistanser på  $120\Omega$ . CAN-bussens längd bestäms efter baud, vilken måste vara samma på alla enheter kopplade till bussen för att de ska kunna kommunicera med varandra (med långsammare baud kan man ha längre CAN-buss). I figuren nedan är ett exempel på hur topologin i ett CAN-nätverk kan se ut. [4] [8]



Figur 4, CAN-nätverk topologi. [8]

Det andra skiktet, data link-skiktet, är länken mellan det fysiska skiktet och nätverksskiktet. Data link-skiktet är uppbyggt av två underskikt, LLC sublayer och MAC sublayer. LLC-skiktet används för att upptäcka fel som kan uppstå i det fysiska skiktet och försöka korrigera felen. MAC-skiktet hanterar åtkomsten till nätverket, så som att förhindra eller hantera kollisioner när många noder är anslutna och vill skriva till nätverket samtidigt. [4] [9]

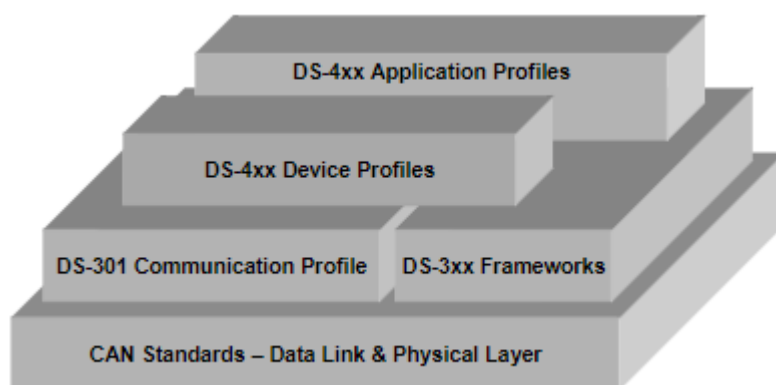
LLC- och MAC-skikten specificeras i standarden ISO-11898 vid användning av CAN-protokoll. För att undvika kollisioner använder CAN-bussen Carrier Sense Multiple Access/Collision Detection som MAC-metod. Den CANopen-nod med högst prioritet har företräde vid kollision medan de andra noderna avslutar överföringen och påbörjar mottagning. Data link-skiktet inom CAN placerar även bitarna i frames. Se figur nedan. [4] [10]



Figur 5, CAN-basformat för data frames. [4]

CAN-data frame börjar med en dominant start bit. Efter detta kommer 11-bit CAN-meddelande identifieraren. I CANopen är detta oftast en del av COB-ID. Eftersom detta fält används för att lösa kollisioner är det viktigt att varje meddelande-ID är unikt i nätverket och att ett CAN-ID kan endast skickas av en nod. De tre följande bitarna, kontrollbitarna, ska beaktas som reserverade och ställas till 0. CiA rekommenderar att inte använda RTR (remote request, används för att ta emot specifikt meddelande) eller IDE (används för att aktivera 29-bit identifierare istället för 11-bit) inom CANopen nätverk. DLC (Data Length Code) specificerar hur många data bytes det finns i ett frame. Tillåtna DLC-värden är bara 0 till 8 fastän DLC är ett 4-bits värde. Datafältet innehåller så många data bytes som DLC specificerar. Vilket betyder längden är 0, 8, 16, 24, 32, 40, 48, 56 eller 64. Efter datafältet kommer CRC (Cyclic Redundancy Checksum). Detta används för att upptäcka fel och ändringar i meddelandet. De resterande kontrollbitarna är CRC delimiter, ACK och ACK delimiter. Delimiter ("skiljetecken") används för att ge noderna lite extra tid att behandla de föregående bitarna. De mottagande noderna har en bit-längd tid (CRC delimiter) att jämföra den beräknade CRC med CRC mottaget i meddelandet. Sedan en bit-längd tid (ACK delimiter) att räkna färdigt CRC och en bit-längd tid (ACK bit) att bekräfta att de har mottagit meddelandet. Om den sista delimitern är 0 betyder det att alla noder har mottagit meddelandet och fått korrekt CRC. Data frame slutar med end-of-frame sekvens med 7 nollor i rad. Beroende på antalet data bytes i ett frame så kan längden variera mellan 44 till 108 bitar. Det måste vara ett uppehåll på minst 2 (enligt ISO-11898-1) eller 3 (enligt ISO-11898) noll-bitar innan nästa frame får börja. [4]

OSI-modellens högsta skikt, applikationsskiktet, är det skikt som interagerar med operativsystem eller programvara. Det viktigaste med CANopen-protokollet är att definiera applikationsskiktet. CANopen-protokollet implementerar även delar av nätverkskiktets funktionalitet till presentationsskiktet. I figuren nedan ser man hur CAN-standarden (ISO-11898) med fysiska- och data link-skiktet och de övre profilerna (DS-301, DS-3xx, DS-4xx) som tillhör CANopen-protokollet bygger på varandra eller kompletterar varandra. [4]



Figur 6, CAN- och CANopen-protokoll [4]

DS-301 (CANopen Application Layer and Communication Profile) är den viktigaste profilen i CANopen-protokollet och alla CANopen-enheter ska implementera CiA DS-301. Denna profil definierar den grundläggande kommunikationen, datastrukturen och nätverkshanteringen i CAN-nätverket. Frameworks är på samma nivå som Communication profilen därmed har de beteckningen DS-3xx. CAN-bussen har i CiA dokumentationen beteckningarna DS-1xx och DS-2xx samt olika EN- och ISO-standarder. DS-3xx serien är kommunikationsprofiler baserade på DS-301. Dessa används för att uppfylla kommunikationskraven i specifika fall. Exempel i tabellen nedan. [4]

Tabell 1, CANopen Frameworks. [4]

Framework	Title
DSP302	CANopen Framework for CANopen Managers and Programmable CANopen Devices
DSP304	CANopen Framework for Safety-relevant Communication
DSP305	CANopen Layer Setting Services and Protocols (LSS)
DSP307	CANopen Framework for Maritime Electronics

Device profiles definierar kommunikationsmedel för en specifik typ av utrustning ansluten till en CAN-buss. I en industriell miljö bör varje typ av enhet, som är kopplad till en CAN-buss, ha en device profile. Tabellen nedan visar ett antal vanliga profiler.

**Tabell 2, Device Profiles. [4]**

<b>Device Profile</b>	<b>Title</b>
DS401	CANopen device profile for generic I/O modules
DSP402	CANopen device profile for drives and motion control
DS404	CANopen device profile for measuring devices and closed loop controllers
DS405	CANopen interface and device profile for IEC 61131-3 programmable devices
DS406	CANopen device profile for encoders
DSP408	CANopen device profile for fluid power technology proportional valves and hydrostatic transmissions
DSP410	CANopen device profile for inclinometers
DSP413	CANopen device profiles for truck gateways

### 2.2.2 DS-301

CiA DS-301 är den viktigaste standarden i CANopen-protokollet eftersom den definierar Object Dictionary (OD), Electronic Data Sheet (EDS), Service Data Object (SDO), Process Data Object (PDO), Network Management (NMT) samt en del funktioner (t.ex. SYNC, TIME, EMCY).

Object Dictionary motsvarar en tabell som innehåller alla nätverksåtkomliga data och varje CANopen-enhet har endast en OD. Alla konfigurationsparametrar och processparametrar för enheten sparas här. Varje post i OD har ett nummer som kallas index, med vilket man kommer åt posten. Index är 16 bitar stort och därmed finns det max 65536 poster. Varje post i OD kan ha upp till 256 underposter som kallas subindex och representeras av ett 8-bitars värde. Varje post har minst en underpost. Med hjälp av index och subindex kommer man åt de specifika parametrar man behöver. Tabell 3 visar hur de 65536 indexen delas upp och vilka typer av data som hör vart. [4]

Tabell 3, Indexering av Object Dictionary. [4]

Index Range	Description
0000h	Reserved
0001h – 0FFFh	Data Types
1000h – 1FFFh	Communication Entries
2000h – 5FFFh	Manufacturer Specific
6000h – 9FFFh	Device Profile Parameters
A000h – FFFFh	Reserved

Electronic Data Sheet används av mjukvara för att specificera vilka OD-poster som stöds av en CANopen-nod. Ett exempel på en OD-post i en EDS-fil kan ses i figuren nedan.

```
[1018sub2]
ParameterName=Product code
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0x03
PDOMapping=0
```

Figur 7, Object Dictionary-post i Electronic Data Sheet-fil. [4]

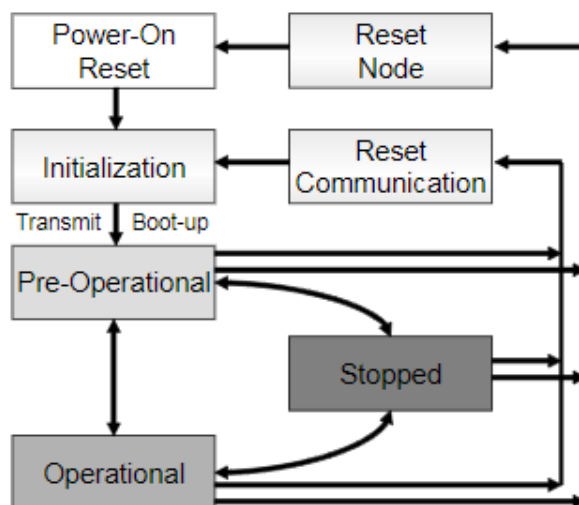
1018sub2 betyder att denna post har subindex 2 i index 0x1018. ParameterName är namnet på OD-posten. I detta exempel är det en produktkod. ObjectType=0x7 betyder att innehållet är en variabel. DataType=0x0007 specificerar att datatypen är en unsigned 32-bit integer. AccessType=ro betyder att objektet är read only, d.v.s man kan endast läsa värdet och inte skriva till posten. DefaultValue=0x03 visar värdet för denna post. PDOMapping=0 betyder att objektet inte stöder PDO mapping. [4]

Service Data Object används för att etablera server/klient-förhållanden mellan CANopen-enheter. Den enhet som stöder SDO kallas SDO-Server. Klientenheten kan komma åt serverenhetens OD genom att läsa eller skriva dess index och subindex. Oftast används endast en masternod som klient i ett CANopen-nätverk för att konfigurera parametrar för andra noder via SDO-meddelanden. Med SDO kan man komma åt alla parametrar men för att använda processparametrar är det mer ändamålsenligt att använda PDO. [4]



Process Data Object används för att skicka data i realtid. Ett PDO som tas emot av en nod kallas Receive PDO (RPDO) medan ett PDO som skickas kallas Transmit PDO (TPDO) och varje PDO har en meddelandeidentifierare (COB-ID). Det finns ett par olika överföringsmetoder, synkroniserad och osynkroniserad. Med synkroniserad överföring skickar CANopen-enheten ett PDO när den tar emot ett synkroniseringsmeddelande (SYNC). Med osynkroniserad överföring är PDO händelsedriven. När det sker en förändring av processdata överförs PDO enligt CANopen-protokollinställningen som implementerats i CANopen-enheten. Det finns fyra olika metoder för att trigga överföring: händelse, timer, fjärbegärd (Remote Transmission Request) och synkroniserad. Händelsedriven överföring sker när som helst som en händelse inträffar i en CANopen-enhet. Timerdriven överföring utlöser även om ingen händelse har inträffat. Fjärbegäran fungerar genom att en enhet kan be en annan enhet skicka ett PDO genom att skicka en fjärbegäran (RTR). [4]

Network Management har två uppgifter. Kontrollera kommunikationstillståndet för noderna i nätverket och nodövervakning med hjälp av "node guarding" och "heartbeat". Node guarding fungerar genom att mastern med jämna mellanrum frågar slavarerna om de är vid liv medan heartbeat fungerar genom att slavarerna kontinuerligt skickar sitt NMT-tillstånd till bussen. Heartbeat är den rekommenderade metoden. I CANopen-nätverk finns endast en NMT master, de andra noderna är NMT slave-noder. NMT mastern kontrollerar kommunikationstillstånden för slavarerna genom att sända kommandon. Det finns fyra olika tillstånd för slavarerna: initialization (initialisering), pre-operational (pre-operativ), operational (operativ) och stopped (stoppad). Figur 8 visar förhållandet mellan de olika NMT-tillstånden för slave-noderna. [4]



Figur 8, Network Management-tillstånden för slave-noder. [4]

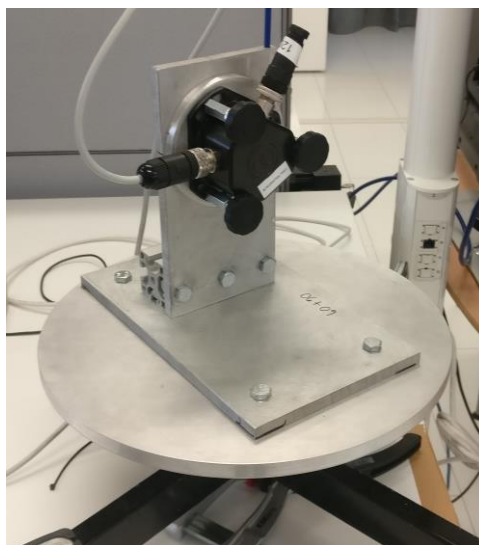
När en slave-nod startas eller återställs intar den initialiseringsstillstånd. Efter initialiseringen skickar noden ett startmeddelande till NMT master-noden och intar automatiskt pre-operativt tillstånd. När en slave-nod är i pre-operativt tillstånd kan master-noden konfigurera parametrar för slaven. Slave-noden intar operativt tillstånd när master-noden ger kommando för detta. Ifall en slave-nod får ett allvarligt kommunikationsfel så försätter master-noden slave-noden i stoppat tillstånd. Den huvudsakliga skillnaden mellan de olika kommunikationstillstånden är att inte alla CANopen-kommunikationstyper är aktiva i varje tillstånd. Tabell 4 visar vilka typer som är aktiva i de olika tillstånden. [4]

**Tabell 4, Kommunikationstyper aktiva vid olika tillstånd. [4]**

	Initializing	Pre-operational	Operational	Stopped
Boot-Up	♦			
SDO		♦	♦	
Emergency		♦	♦	
SYNC/TIME		♦	♦	
Heartbeat/ Nodeguard		♦	♦	♦
PDO			♦	

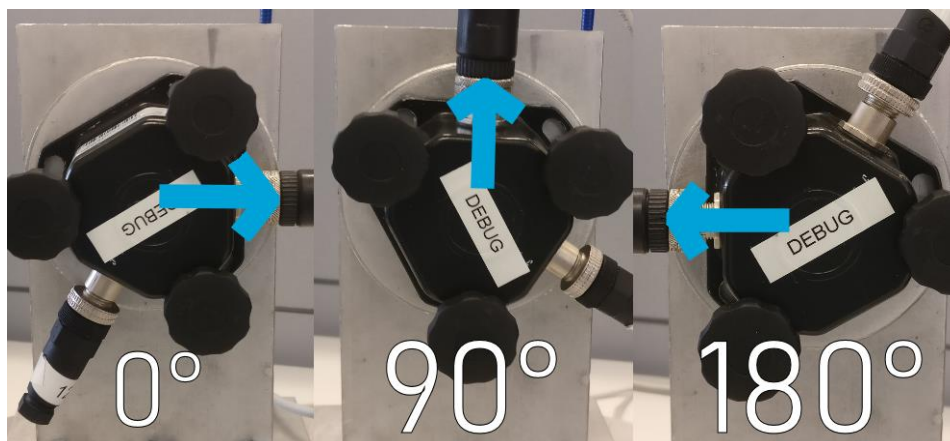
## 2.3 Testprocess

Detta kapitel beskriver hur AMU-GEO:s noggrannhet kontrolleras. Automatiseringen av detta har varit huvuduppgiften i examensarbetet. Till testet används en rotationsjigg och ett CAN-analyseringsprogram (CANtrace).



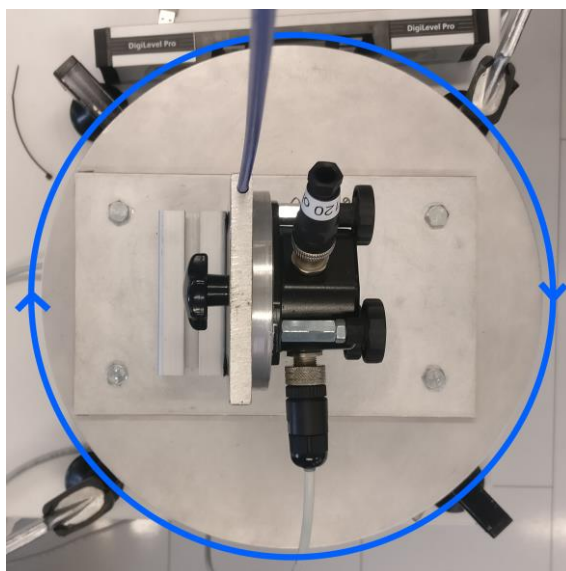
**Figur 9, Testjigg.**

AMU-GEO fästs vertikalt i rotationsjiggen och ska testas i tre lägen,  $0^\circ$ ,  $90^\circ$  och  $180^\circ$  (se figur 10). Med hjälp av CANtrace ser man till att vinkeln (angle360) är  $\pm 0.10^\circ$  inom gränsen.



**Figur 10, AMU-GEO i tre olika lägen för test.**

För varje vinkel ska jiggen roteras medsols  $360^\circ$  med jämn fart på 5 – 10 sekunder. Innan man roterar måste jiggen vara helt stilla så att AMU-GEO:s mätvärden inte påverkas av störningar och så att startvärdet (kallad GyroAngleZ i AMU-GEO) är så noga som möjligt. När jiggen står stilla ska man spara undan startvärdet.



**Figur 11, Testjigg roteras medsols.**

Efter man roterat jiggen  $360^\circ$  ska jiggen stanna helt innan man sparar undan slutvärdet. När alla 3 test är slutförda ska resultatvärden fyllas i ett kalkylark som beräknar felen och visar

om sensorn blivit godkänd eller ej. Figuren nedan visar CANtrace-fönstret där man läser av Angle360, AngVels (vinkelhastigheterna) och GyroAngleZ.

Id	Ch	Dlc	Data	Δt	Count	Dir	Name	Interpretation
181	1	4	01 00 03 00	0.0520	2071	Rx	AMUGEO360_AN	
			Valid		3			
381	1	6	00 00 00 00 00 00	0.0520	2071	Rx	AMUGEO360_AN	
			Angle360		0.01		deg	
			AngVel_X		0		deg/s	
			AngVel_Y		0		deg/s	
			AngVel_Z		0		deg/s	
481	1	8	f3 b9 68 16 d9 03 bb 00	0.0520	2071	Rx	AMUGEO360_GY	
			Temperature		18.7		C	
			GyroAngleX		-179.33		deg	
			GyroAngleY		57.36		deg	
			GyroAngleZ		9.85		deg	
701	1	1	05	1.0070	113	Rx		Node: 0x1 Operati

Figur 12, CANtrace-fönster.

### 2.3.1 Exempeltest

Tabellen nedan visar mätvärdena från ett rotationstest.

Tabell 5, Start- och slutvärden från rotationstest.

	Angle360	GyroAngleZ start	GyroangleZ end
Test case 1	0	1,51	1,47
Test case 2	90	91,86	91,65
Test case 3	180	-178,87	-178,88

För att beräkna om sensorn är godkänd beräknas först största differensen av mätvärdena och sedan det teoretiska maxfelet. I detta fall är största differensen 0,21 och det teoretiska maxfelet är 0,21 (beräknat med formeln nedan, TC DIFF betyder skillnaden mellan start och slutvärde för testet).

$$\text{Teoretiskt maxfel} = \left| \frac{|TC1 DIFF|}{\sin(\tan^{-1}(\frac{|TC1 DIFF|}{|TC2 DIFF|}))} \right|$$

För att en sensor ska godkännas måste maxfelet (det större av de två) vara mindre än 1,70 och differensen mellan maxfelet och det teoretiska maxfelet vara mindre än 0,20. I detta exempel är den testade sensorn godkänd.

## 3 Utvecklingsmiljö

I detta kapitel beskrivs miljöerna och språken som testprogrammet och konverteringsskriptet utvecklats i.

### 3.1 Microsoft Visual Studio

Microsoft Visual Studio är en avancerad integrerad utvecklingsmiljö som används för att skapa konsol-, webb- och mobilapplikationer, grafiska användargränssnitt samt moln- och webbtjänster. Med hjälp av denna utvecklingsmiljö kan man skapa och behandla kod och projekt. Visual Studio stöder de flesta allmänna programspråk och i detta arbete användes C++ och Python. Detta program användes för att utveckla det huvudsakliga testprogrammet i examensarbetet. [11]

### 3.2 Notepad++

Notepad++ är en kodredigerare (enklare utvecklingsmiljö än Visual Studio) som stöder omkring 80 programspråk. Detta program används oftast för att skriva och redigera enskilda filer för program och projekt. I examensarbetet användes detta program för att utveckla skriptet som konverterar resultatfilen till ett kalkylark. [12]

## 3.3 Programspråk

Programspråk, eller programmeringsspråk, är språk som används inom datorprogrammering för att skapa funktioner och algoritmer.

### 3.3.1 C++

C++ är ett programspråk, baserat på språket C, som används för avancerade program och ger programmerare stor kontroll över systemresurser och minnen. Språket är objektorienterat vilket ger en klar struktur över programmet och tillåter programmerare att återanvända samma kod till andra program. [13]

### 3.3.2 Python

Python är ett avancerat, universalprogramspråk som ofta används för integrerade system och datorprogram som ofta inte har användargränssnitt såsom skript. Liksom C++ är Python

objektorienterat vilket gör programkoden lättläslig och språkets syntax gör det möjligt att skapa fungerande program med färre kodrader. [14]

### 3.3.3 JSON

JSON (JavaScript Object Notation) är ett dataformat som är läs- och skrivbart för människor. Datorprogram kan enkelt tyda och generera JSON-filer. Textformatet är helt oberoende av programspråk men använder liknande regler som C, C++, JavaScript, Python etc. Detta gör JSON till ett ideellt språk för datautbyte. [15]

## 4 Utförande

Detta kapitel förklarar hur planeringen och utvecklingen av programmet utfördes.

### 4.1 Krav

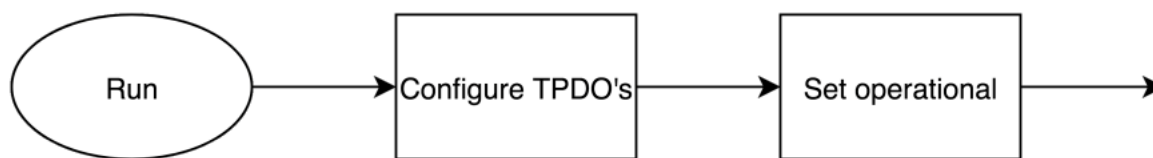
Uppgiften var att automatisera testprocessen av inklinometer och gyroskop, AMU-GEO 360, för att kontrollera sensorns noggrannhet och säkerställa att sensorn har blivit korrekt kalibrerad av tillverkaren. Programmet ska konfigurera sensorn, läsa startvärden från sensorn och spara undan dem, be användaren rotera testjiggen enligt anvisningarna, läsa och spara slutvärden, beräkna resultatvärden och skriva ut resultaten i både programfönstret och i en resultatfil. Resultatfilerna från testen av sensorerna i en batch ska sedan konverteras till ett kalkylark som visar vilka sensorer som fått godkänt resultat och sensorernas noggrannheter.

### 4.2 Planering

Planeringen av arbetet börjades med att skapa grundläggande flödesscheman för hur programmet och skriptet ska fungera och en lista med nödvändiga variabler som behövdes. Flödesschemat för huvudprogrammet kommer tas upp i delar i följande kapitel. Se bilaga för det fullständiga slutgiltiga flödesschemat.

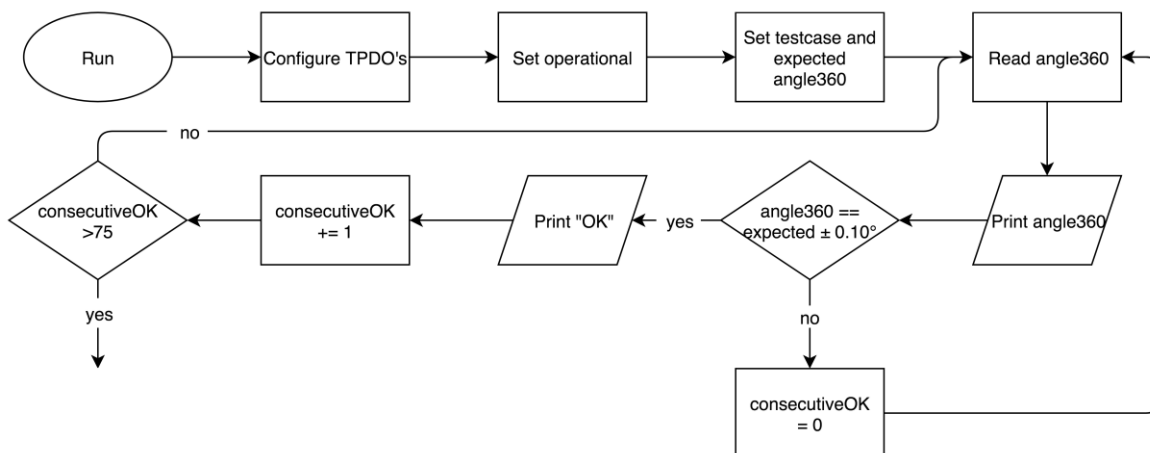
#### 4.2.1 Huvudprogram

Det första programmet måste göra är att ställa in kommunikationsinställningarna, vilka PDO:s (Process Data Object) som AMU-GEO ska skicka samt sätta AMU-GEO i operativt läge.



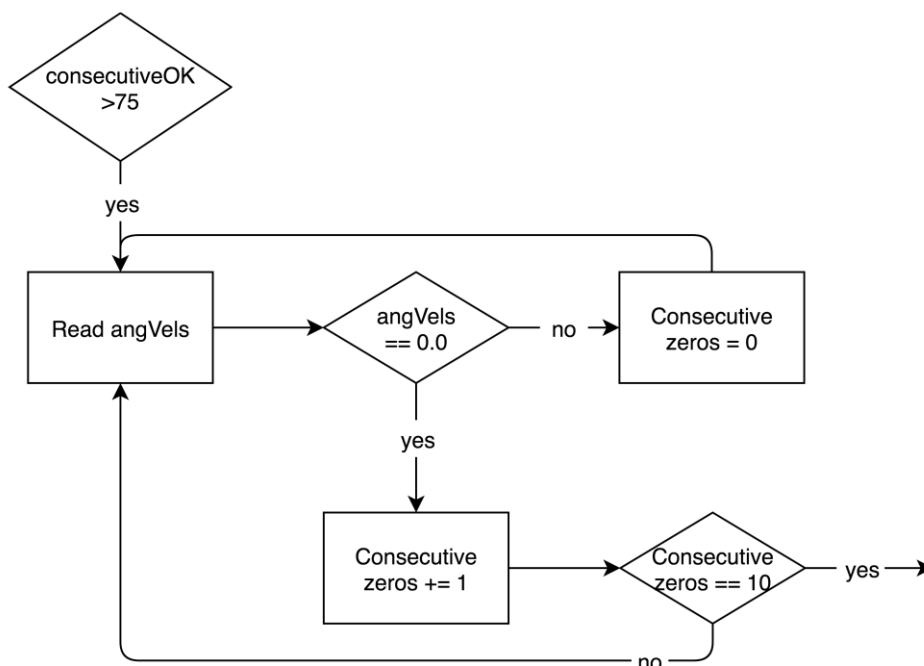
Figur 13, Flödesschema del 1 (initialisering av AMU-GEO).

När AMU-GEO är färdigt konfigurerad påbörjas testet. Testet ska börja med att AMU-GEO:s vinkel (parameternamn angle360) ska sättas i  $0^\circ$  (enligt figur 10, kapitel 2.3). Programmet ska kontinuerligt kontrollera att angle360 är  $\pm 0.10^\circ$  inom gränsen. För att se till att angle360 inte svajar och verkligen är inom gränsen ska minst 75 avläsningar i rad vara godkända.



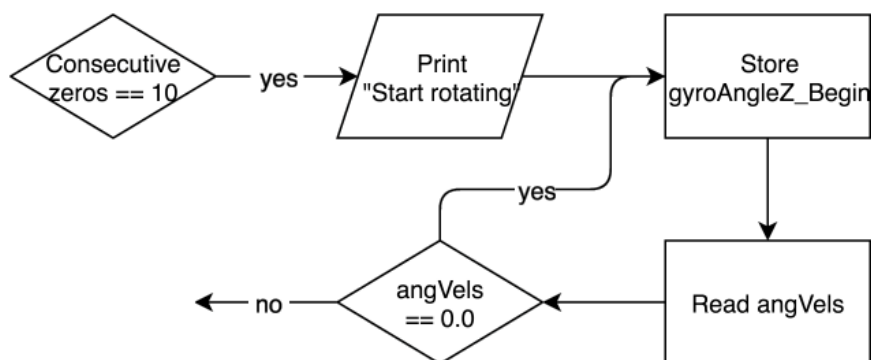
Figur 14, Flödesschema del 2 (kontroll av angle360 (AMU-GEO:s vinkel)).

Efter säkerställningen att AMU-GEO:s vinkel är korrekt ska programmet se till att sensorns vinkelhastigheter (angVels) är 0 innan rotering av jiggen kan påbörjas för att få så noggrant startvärde som möjligt. Vinkelhastigheter ska avläsas som 0 tio gånger i rad innan startvärdet kan avläsas.



Figur 15, Flödesschema del 3 (kontroll av vinkelhastighet).

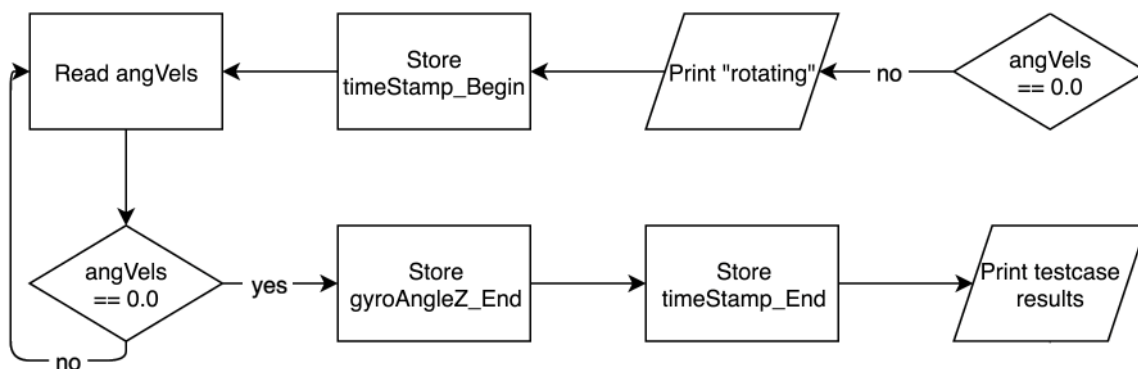
När AMU-GEO:s vinkelhastighet har hållits vid 0 i tio avläsningar ber programmet användaren att påbörja rotering av testjiggen medan det kontinuerligt sparar undan startvärdet (gyroAngleZ) tills AMU-GEO känner av rörelse. När sensorn börjar röra på sig behåller programmet det senaste mätvärdet som startvärde (när sensorn var stillastående).



Figur 16, Flödesschema del 4 (sparande av startvärde innan rotering).

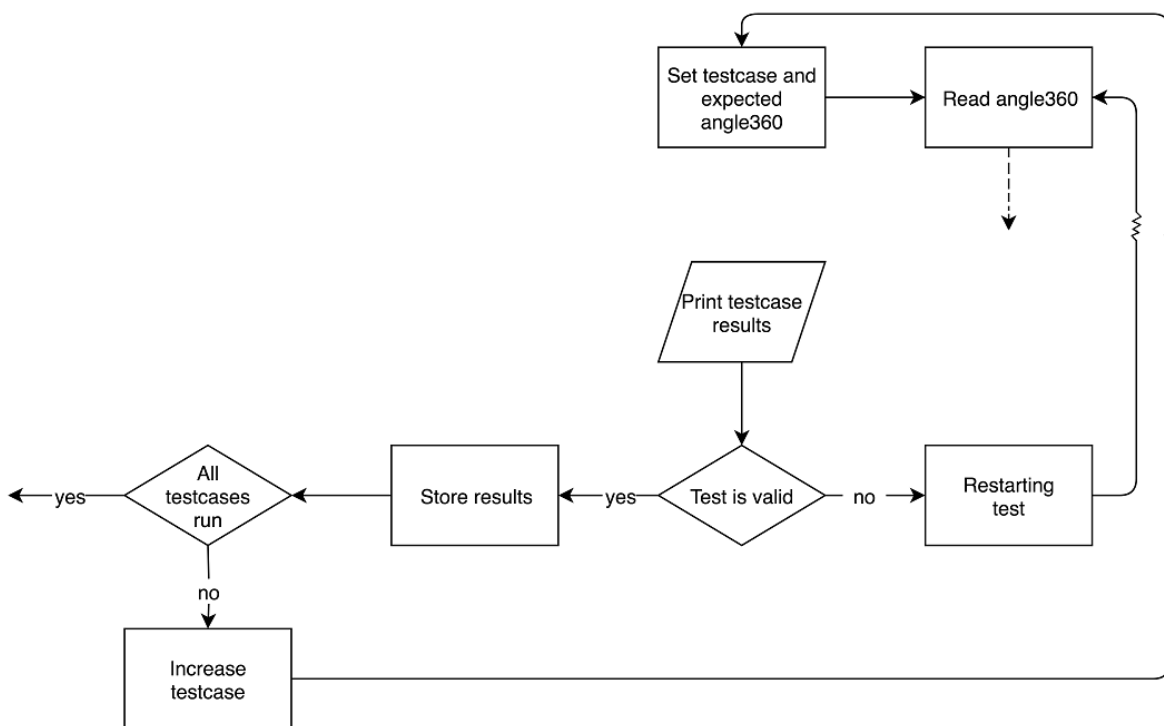
När AMU-GEO börjar röra sig skriver programmet ut att sensorn roterar och sparar undan en tidsstämpel och går in i en loop som fortsätter tills sensorn stannat helt (vinkelhastighet är 0). Slutvärdet och en till tidsstämpel sparas undan och resultat för deltestet skrivs ut i konsolfönstret.





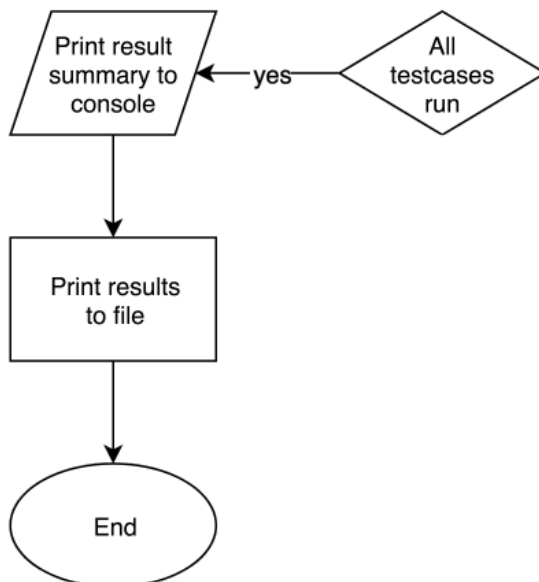
Figur 17, Flödesschema del 5.

I konsolfönstret ser användaren tiden för hur länge jiggen roterats, start- och slutvärdena för gyroAngleZ samt skillnaden mellan dessa. Om användaren roterar jiggen för fort (mindre än 5 sekunder) eller för långsamt (mer än 10 sekunder) blir testet underkänt och programmet startar om testet. Ifall användaren rycker för hårt i jiggen vid rotering kommer AMU-GEO:s vinkelhastighet bli för hög vilket också kommer resultera i att testet startas om. Om testet godkänns sparas mätvärdena undan och programmet kontrollerar ifall alla tre test blivit utförda (0°, 90° och 180°). Om alla tre test inte är utförda går programmet vidare till nästa test och utför samma process för de resterande vinklarna. Ifall alla tre test är genomförda går programmet vidare till slutsteget.



Figur 18, Flödesschema del 6.

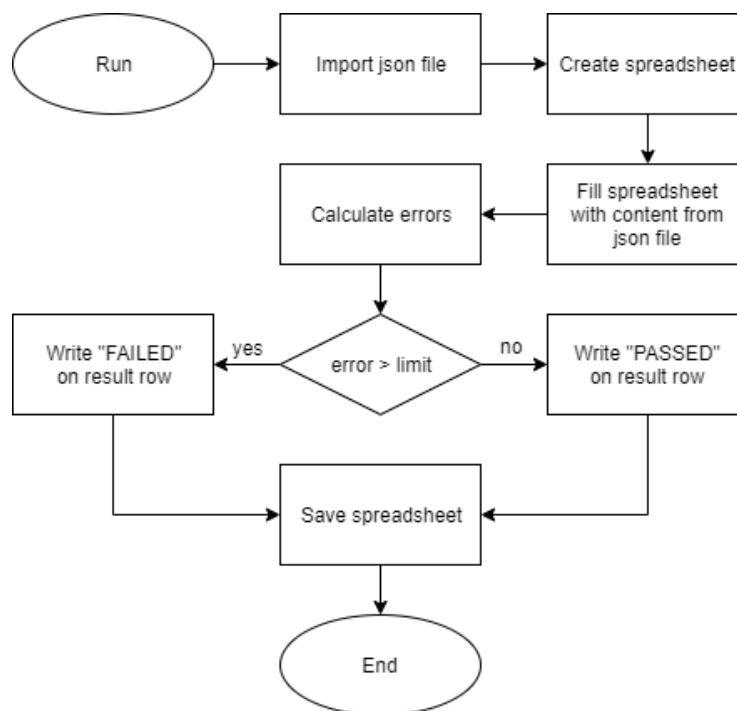
I slutskedet skriver programmet ut en sammanfattning av testen, de beräknade maxfelen (förklaras i kapitel 2.3.1) och om den testade AMU-GEO:n godkänns eller ej baserat på de beräknade värdena. Dessa värden sparas även undan i en JSON-fil som med hjälp av ett konverteringsskript skrivs om till ett kalkylark.



Figur 19, Flödesschema del 7 (Utskrivning samt sparande av testresultat).

#### 4.2.2 Konverteringsskript

Skriptet börjar med att importera resultatfilerna. Alla JSON-filer som finns i samma mapp som skriptet kommer med i kalkylarket. Ett kalkylark skapas och celler fylls med nödvändig information (serienummer, parameternamn etc.) för att tyda resultaten. Testresultaten från filerna fylls sedan radvis och felen beräknas (formel för detta finns i kapitel 2.3.1). Om dessa fel är innanför de godkända gränserna skrivs "PASSED" i slutet av raden för att visa att den sensorn blivit godkänd. Men om felet är större än gränsvärdet kommer hela raden markeras med rött och i slutet av raden skrivs "FAILED" för att enkelt se vilka sensorer i en batch som blivit underkända. Efter att skriptet gått igenom varje fil i mappen kommer kalkylarket sparas och skriptet avslutas.



Figur 20, Flödesschema för konverteringsskript.

## 5 Resultat

Resultatet blev ett lättanvänt program som automatiserade de mest tidskrävande delarna av testprocessen för AMU-GEO samt ett skript som konverterar resultatfilerna från en batch till ett kalkylark som visar alla resultat och vilka sensorer som blivit godkända. Förut kunde testandet av en AMU-GEO ta mer än 5 minuter på grund av manuell inmatning av data men eftersom programmet numera automatiserar allt detta tar ett problemfritt test endast 1–2 minuter.

### 5.1 Testprogram

När programmet startas ser användaren om programmet kommunicerar med AMU-GEO och om alla funktioner utförts korrekt. Programmet ber sedan användaren välja om testet ska startas, sensorn ska återställas till fabriksinställningar eller om programmet ska avslutas.

```

*****
      Configuration
*****
Executing function readVersion:
Done.
Executing function checkAmuGeo360AutoCalibration:
Checking if autocalibration is active, timeout 90 seconds. Please wait...
Done.

*****
      Rotation Test
*****
0   Exit
1   Run test
9   Restore factory default settings and exit
*****
Please type a number <0 to Quit>:

```

Figur 21, Startmeny.

I konsolfönstret skrivs vilket test som körs och i vilken vinkel sensorn ska sättas. Med hjälp av indikatorn i figuren nedan ser användaren hur långt sensorn är från den korrekta vinkeln. Om den vertikala linjen är på eller mellan punkterna så är  $\text{angle360} \pm 0.10^\circ$  inom gränsen och testet kan fortsätta.

```

*****
      TC-2: Set Angle360 to 90 degrees.
*****
Angle360 = 88.83      ____!____. . . . .
*****

*****
      TC-2: Set Angle360 to 90 degrees.
*****
Angle360 = -0.06     <____. . . . .
*****

*****
      TC-3: Set Angle360 to 180 degrees.
*****
Angle360 = -176.26  _____!____
*****

*****
      TC-2: Set Angle360 to 90 degrees.
*****
Angle360 = 90.00     OK_____!____

```

Figur 22, angle360 indikator.

När ett test är färdigt skrivs en mellanrapport (figur 23) ut som visar tiden, start- och slutvärdena samt differensen mellan dem. Efter detta startas nästa test automatiskt och användaren besätta AMU-GEO i korrekt vinkel.

```

*****
TC-1: Set Angle360 to 0 degrees.
*****
Angle360 = -0.02      OK_____! .._____
Start rotating.
Rotating...

Time: 6.89 seconds.
GyroAngleZ_Begin = 67.58
GyroAngleZ_End = 67.58
!Difference! = 0.00

```

Figur 23, Mellanrapport med resultat från deltest 1.

Om användaren inte roterar jiggen med jämn fart kommer programmet registrera en snabb ökning av hastigheten och testet kommer misslyckas (figur 24). I detta fall startas samma test automatiskt om igen.

```

*****
TC-3: Set Angle360 to 180 degrees.
*****
Angle360 = -180.00   OK_____! .._____
Start rotating.
Rotating...

Time: 6.03 seconds.
GyroAngleZ_Begin = -74.94
GyroAngleZ_End = -75.11
!Difference! = 0.17
TEST FAILED (HIGH ANGUEL), restarting test.

```

Figur 24, Misslyckat testförsök.

Programmet beräknar slutligen felen och skriver ut en sammanfattning av resultaten och om sensorn blivit godkänd eller ej.

```

*****
Results:
*****
TC-1 (0 degrees):
GyroAngleZ_Begin = 19.21
GyroAngleZ_End = 19.37
!Difference! = 0.16

TC-2 (90 degrees):
GyroAngleZ_Begin = 109.86
GyroAngleZ_End = 109.78
!Difference! = 0.08

TC-3 (180 degrees):
GyroAngleZ_Begin = -160.90
GyroAngleZ_End = -160.87
!Difference! = 0.03

Max !Diff! = 0.16
Theoretical !maxerror! = 0.18
Max !error! = 0.18
Max !error! - Theoretical !maxerr! = 0.00

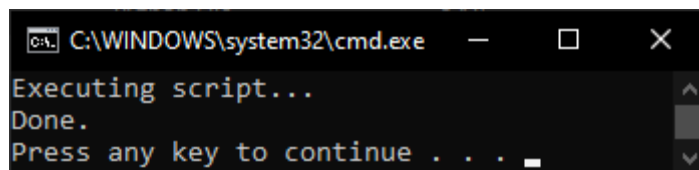
Result: PASSED

```

Figur 25, Resultat från ett komplett test.

## 5.2 Konverteringskript

Genom att starta skriptet med alla resultatfiler i samma mapp skapas ett lättläst kalkylark där man ser vilka sensorer som blivit godkända och vad de har haft för mätvärden.



Figur 26, Skriptet skapar kalkylark.

Alla godkända sensorer märks med "PASSED" i slutet av sin rad och de som inte klarar testet märks med "FAILED" och hela raden blir röd. I exemplet nedan (figur 27, 28) ser man att AMU-GEO 360 med serienummer 16057700572 har blivit underkänd eftersom maxfelet överskrider gränsen på 1.70°.

	A	B	C	D	E	F	G	H	I
1	Autogenerated on: 2019-05-29								
2	AMU-GEO 360 - Rotation Test Specification, rev: XXXX								
3	Firmware version: X.X.X.X								
4	Passing criteria:								
5	Max  error  <= 1.70 (1.50 + delta)								
6	Max  error  - Theoretical  maxerr  <= 0.20 (tolerance added for random drift)								
7									
8	Angle360	0°				90°			
9	Serial	GyroAngleZ @ 0°	GyroAngleZ @ 360°	Diff	Diff	GyroAngleZ @ 0°	GyroAngleZ @ 360°	Diff	Diff
10	19097700050	1,51	1,47	-0,04	0,04	91,86	91,65	-0,21	0,21
11	16057700572	76,01	76,03	0,02	0,02	138,19	136,26	-1,93	1,93
12	18397700202	-9,04	-8,51	0,53	0,53	81,76	81,91	0,15	0,15
13	1814770018B	6,53	6,76	0,23	0,23	36,7	36,86	0,16	0,16

Figur 27, Kalkylark med resultat från 4 sensorer.

J	K	L	M	N	O	P	Q	R
180°								
GyroAngleZ @ 0°	GyroAngleZ @ 360°	Diff	Diff	Max  diff	Theoretical  maxerr	Max  error	Max  error  - Theoretical  maxerr	Result
-178,87	-178,88	-0,01	0,01	0,21	0,21	0,21	0,00	PASSED
-138,23	-137,81	0,42	0,42	1,93	1,93	1,93	0,00	FAILED
171,28	170,95	-0,3	0,33	0,53	0,55	0,55	0,00	PASSED
127,53	127,05	-0,5	0,48	0,48	0,28	0,48	0,20	PASSED

Figur 28, Kalkylark med resultat från 4 sensorer (forts.).

## 6 Diskussion

Examensarbetet har varit väldigt lärorikt eftersom jag inte hade någon tidigare kunskap om CAN-bussar och jag inte förut hade programmerat något så avancerat. Både jag och TK Engineering blev nöjda med slutresultatet men det finns förstås en del saker som kan förbättras med testprocessen, t.ex. installera en motor i testjiggen som skulle få bort den mänskliga faktorn, rotera jiggen med jämn fart och ge exaktare testresultat.

Som tidigare nämnt blev testprocessen mycket snabbare och TK Engineering behöver längre inte ha en anställd spendera en hel arbetsdag på att testa en batch sensorer och sparar därmed både tid och pengar.

## 7 Källförteckning

- [1] ”Finder,” [Online]. Available: <https://www.finder.fi/Elektroniikkalaitteistot/TK+Engineering/Vaasa/yhteystiedot/465690>. [Använd September 2019].
- [2] ”TKE Website,” TK Engineering, [Online]. Available: <https://tke.fi/>. [Använd September 2019].
- [3] ”AMU-GEO 360 Angle Sensor - CANopen,” [Online]. Available: <https://tke.fi/site/assets/files/1649/tke-amu-geo-360.pdf>. [Använd Oktober 2019].
- [4] O. Pfeiffer, A. Ayre och C. Keydel, Embedded Networking with CAN and CANopen, Copperhill Technologies Corporation, 2003.
- [5] ”CiA About,” [Online]. Available: <https://www.can-cia.org/about-us/>. [Använd 2019 Oktober].
- [6] ”CANopen.us,” [Online]. Available: <https://www.canopen.us/>. [Använd Oktober 2019].
- [7] ”ITU,” [Online]. Available: [http://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-X.200-199407-I!!PDF-E](http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.200-199407-I!!PDF-E). [Använd Oktober 2019].
- [8] W. Voss, ”Copperhill technologies,” [Online]. Available: <https://copperhilltech.com/blog/controller-area-network-can-bus-physical-layer-and-bus-topology/>. [Använd Oktober 2019].

- [9] P. Zhang, Industrial Control Technology, William Andrew, 2008.
- [10] "CiA CAN data link layer in some detail," [Online]. Available: <https://www.can-cia.org/can-knowledge/can/can-data-link-layers/>. [Använd Oktober 2019].
- [11] "Visual Studio," Microsoft, [Online]. Available: <https://visualstudio.microsoft.com>. [Använd Oktober 2019].
- [12] D. Ho, "Notepad++," [Online]. Available: <https://notepad-plus-plus.org>. [Använd Oktober 2019].
- [13] "cplusplus," [Online]. Available: <http://www.cplusplus.com/info/description/>. [Använd Oktober 2019].
- [14] "Python," [Online]. Available: <https://www.python.org>. [Använd Oktober 2019].
- [15] "JSON website," [Online]. Available: <https://www.json.org/>. [Använd November 2019].



### Fullständigt flödesschema för huvudprogrammet.

