CAN DATA LOGGER FOR X-RAY COLLIMATOR

(NON-PAT INTERNSHIP – SIEMENS HEALTHCARE PVT. LTD.)

A PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

by

K MANOJ MADHAV (15BEC1078)

Under the Guidance of

Dr. SOFANA REKA S

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING
VELLORE INSTITUTE OF TECHNOLOGY
CHENNAI - 600127

April 2019

CERTIFICATE

This is to certify that the Project work titled "***CAN Data Logger for X-Ray Collimator***" that is being submitted by ***K Manoj Madhav (15BEC1078)*** is in partial fulfillment of the requirements for the award of **Bachelor of Technology in Electronics and Communication Engineering**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Sofana Reka S **Internal Guide**	**Ms. Shamlin Sunny** **External Guide - SHPL**

The Project Report is satisfactory / unsatisfactory

Internal Examiner	**External Examiner**

Approved by

PROGRAM CHAIR B. Tech. (ECE)	**DEAN** School of Electronics Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

12/10/2018
Chennai

To

Ms. Shamlin Sunny
Innovation Think Tank Manager
Siemens Healthcare Pvt. Ltd., Bangalore
Gold Hill Supreme Software Park Plot No. 21, 22, 27, 28, Shanthipura Road Electronic City,
Phase II-Industrial Area, Hosur Road Bangalore – 560 100

Sub: Internship-reg

Dear Sir,

Warm greetings from VIT!

VIT University has made a mark in the field of higher education in India, over the past 30 years, by imparting quality education in a cross-cultural ambience combined with extensive application oriented learning process. We strongly believe in having a close and mutually beneficial association with the industries in order to promote experiential learning and expose various project practices followed in industries. In this connection, the students are encouraged to take up in-plant training with leading institutions / industries / research organizations.

As part of our B.Tech curriculum, the student has to undergo in internship training during the vacation period. In this regard, may I request you to kindly permit Mr. K Manoj Madhav (15BEC1078) who is in year 4 of B.Tech ECE in our university to do in-plant training at your concern for a period of 5 Months (Tentative) starting from 07/01/2019. Your assistance in arranging this internship training will be highly appreciated.

Thanking You.
Yours sincerely,

R. Menaka

Dr. MENAKA.R
Associate Professor & Program Chair – B. Tech. ECE
School of Electronics Engineering
Ph: +91 44 3993 1215 Mob: +91 98409 93204
Email. Id: pcelectronicsug.cc@vit.ac.in

VIT UNIVERSITY School of Electronics Engineering Chennai-600 127, Tamil Nadu, India

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt gratitude and regards to my guide, **Dr. Sofana Reka S**, for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The guidance, help and blessings given by her time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to **Ms. Shamlin Sunny, Technology Professional, Siemens Healthcare Pvt. Ltd.**, for her cordial support, valuable information and for trusting me with this project, which helped me in completing this task through various stages.

I am obliged to **Mr. Pradeep P A** and **Mr. Mathew Abraham Chandy**, **Advanced Therapy (AT), Siemens Healthcare Pvt. Ltd.**, for the valuable information provided by them in their respective fields. I am grateful for their cooperation, valuable inputs and constructive criticism provided throughout the completion of the project.

I am extremely grateful to **Dr. Siva Subramanian, Dean of the School of Electronics Engineering, VIT Chennai,** for extending the facilities of the School towards our project and for his unstinting support.

I would like to express my gratitude to the **Programme Chair, Dr. Vetrivelan. P** for his support throughout the course of this project.

Lastly, I thank almighty, my parents, brother and friends for their constant support and encouragement without which this project would not be possible.

K MANOJ MADHAV
15BEC1078

ABSTRACT

This thesis devises a plausible solution to the issue of resolving or viewing CAN Data related issues by means of a technician visiting the client site and plugging in a real time CAN analyzer. The idea is to make a log dump of all the CAN Data being transmitted over the CAN BUS line in the Artis System. The use of this CAN Data Logger is to make a data log of all the CAN Data bitstreams being transmitted in accordance to the communication standards of CAN2.0b.

This CAN Data Logger is the minimalistic hardware addition to any CAN Communication system having a 9-pin DB9 Connector (RS232). The data is tapped from in between the communication line and then data packets in the form of data bitstreams are transmitted serially to a Linux or QNX Operating System which can then be pushed into the Siemens Equipment Backend Server.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

CAN or Controller Area Network is a two wired high-speed serial network communication technology. It is basically used in communications happening among different devices in a small closed region, such as in an automobile or inside medical equipment. It was developed by Robert Bosch in 1982 and released officially by the Detroit's society of Automotive Engineers in 1986. The first car integrating CAN bus was manufactured by Mercedes Benz in 1992.

In this project we are trying to visualize CAN data being communicated over a CAN Bus present between the controller of an X-Ray Collimator and the rotatory motors of the C-Arm, from the client site to any controller/backend terminal at a Siemens Manufacturing or R&D Laboratory.

1.1 Objectives

The following are the objectives of this project:

- To replicate the internal functional controls of an x-ray collimator with minimal hardware peripherals.
- To view the data bitstream on a testing system through a purely software or a minimalistic hardware solution.
- To control the internal rotatory motors (servo) of a C-Arm.
- To visualize the CAN Data stream obtained from over the CAN Bus.
- To push the data obtained into the backend Siemens server for being viewed from any place.
- To device a Minimum Viable Product (MVP) that is retrofittable to the existing ARTIS x-ray collimator system.
- To create a data log of the CAN Data bitstream being communicated over the CAN Bus line for future references and error detections.

1.2 Background and Literature Survey

The problem statement that needed to be addressed in this case was that whenever a problem related to CAN-Bus data transmission or an error in the reception of commands at the actuator side of the collimator cropped up, it demanded the physical presence of a Siemens technician or a service engineer to manually connect a CAN-Bus Analyzer or a CANalyzer to the CAN-Bus line present in between the controller and the actuator of the X-Ray Collimator. This CANalyzer assisted in the viewing of the bitstream being transmitted over the CAN-Bus line without causing any hindrance or data manipulation of the data flowing through the CAN line.

The main objective of this project is to develop a purely software or a minimalistic hardware solution that can be incorporated into the existing system of the Artis X-Ray Collimator product.

Table 1.1 Literature Survey

Sl. No.	Name of article	Name of Conference	Reference
1.	An Embedded CAN-BUS Communication Module for Measurement and Control System.	2010 International Conference on E-Product E-Service and E-Entertainment	The ECM (Embedded CAN Module), its working principles, hardware and software implementations and its performance.
2.	Embedded On-Board Diagnostics System Using CAN Protocol.	2014 International Conference on Communication and Signal Processing	Embedded on-board diagnostics system consists of master slave communication implemented through CAN network.
3.	Supporting Security Protocols on CAN-Based Networks.	2017 IEEE International Conference on Industrial Technology (ICIT)	Arbitrary binary Boolean functions for security purposes.

1.3 Need for this Solution

The need for a solution is to eliminate the necessity for a technician's physical presence at the client location to visualize even minor problems that can be rectified with a minor patch update through software.

This solution shall also create a data log of the CAN Data stream being transmitted across the client and the server system present inside the x-ray collimator generator of the Artis system.

1.4 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2: This chapter describes the design (replication of Artis System), methodology, block diagram and communication standards.
- Chapter 3: This chapter describes the Algorithm and the code flow of the Artis System controller and the collimator.
- Chapter 4: This chapter lists out the components involved and their respective costs. The cost efficiency is also clearly stated.
- Chapter 5: This chapter portrays the result and the output screenshots of this thesis.
- Chapter 6: This chapter denotes the conclusion and future work implementation of this project.
- Chapter 7: This chapter collates all the references and journals that have been referred.

CHAPTER 2

MINIMUM VIABLE PRODUCT FOR THE PROOF OF CONCEPT(PoC)

This Chapter describes the design (replication of Artis System), methodology, block diagram and communication standards.

2.1 Methodology

The idea implemented in this project is to sniff out the CAN Data bitstream from the CAN-Bus line present in between the client and the server of the Artis system. But in order to prove the functionality of this solution, a proof of concept has to be deduced on to a minimal or a scaled down version of the actual product.

Even though the proof of concept is implemented on a very small scale in comparison to the actual product, the communication standards utilized in the proof of concept is in concurrence with the CAN Communication standards with the Artis Product, i.e. CAN 2.0b.

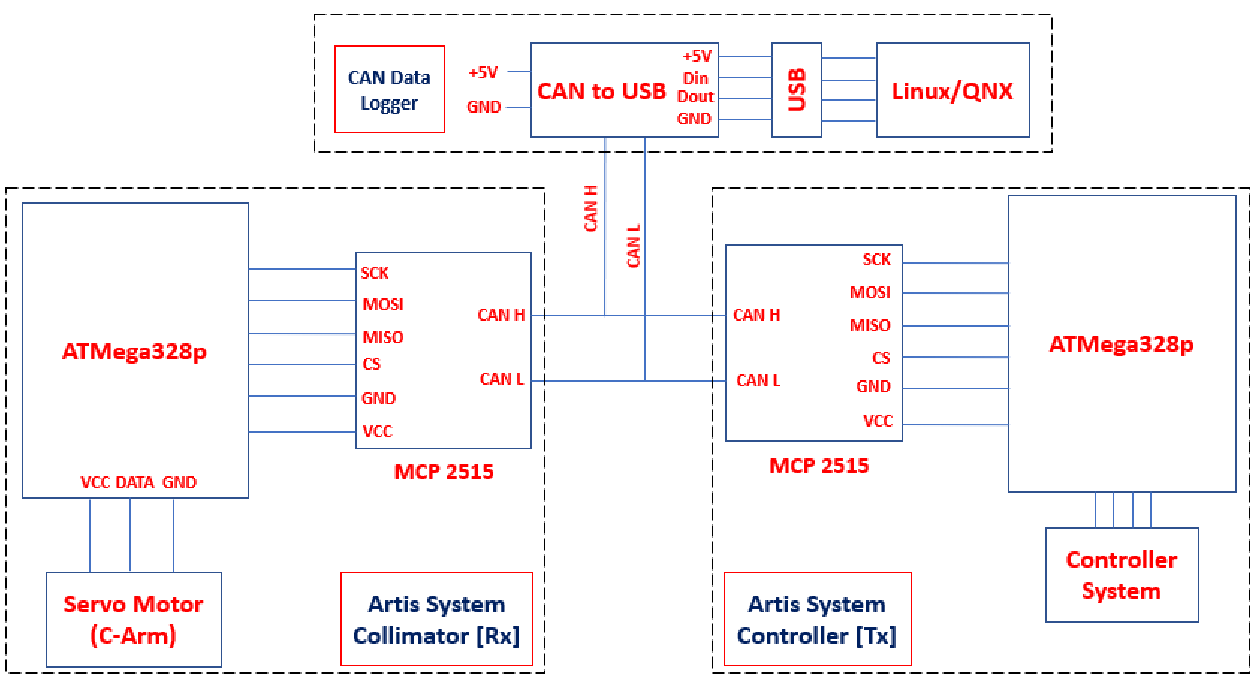


Figure2.1.1 Block diagram of Proof of Concept – Replication of Artis System

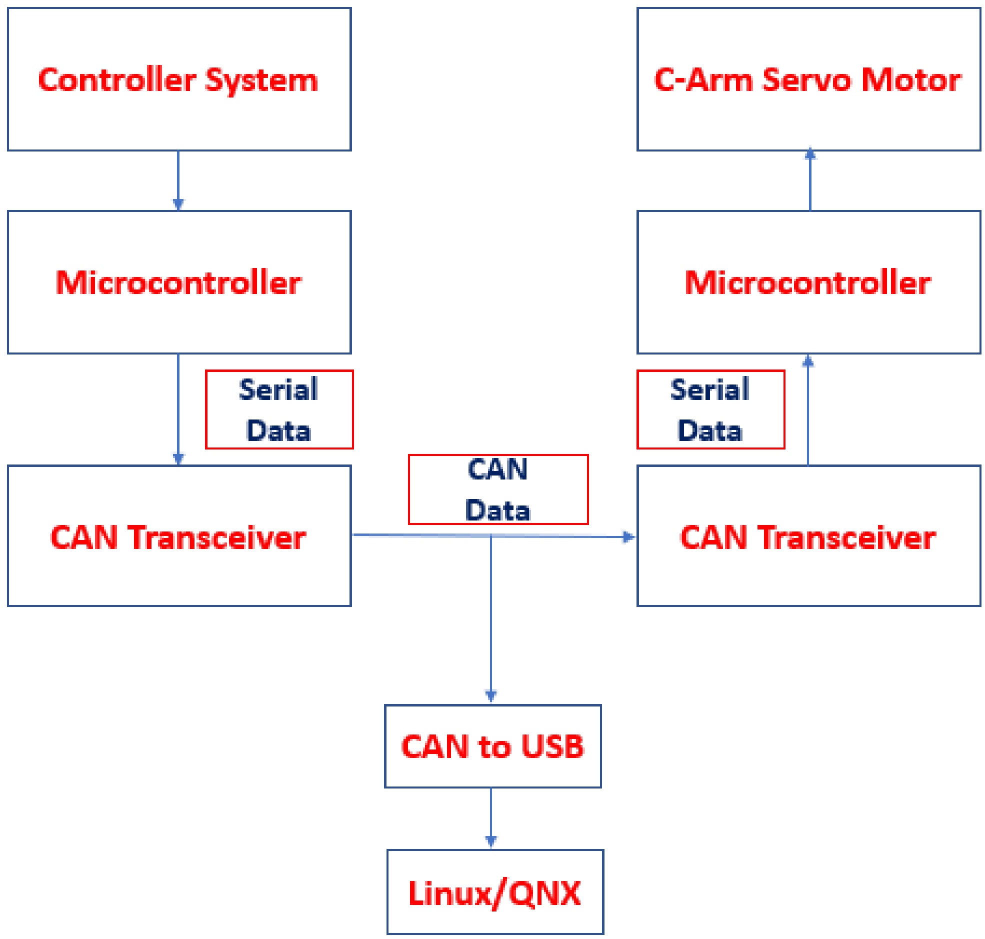


Figure 2.1.2 Flow of Data Bitstream and Control Signals

2.2 Design Approach

The Artis system consists of the above configuration. The entire system can be split into two sections; System Controller [Transmitter –Tx] and System Collimator [Receiver – Rx].

The **System controller [Transmitter –Tx]** section consists of three major components:

- **Controlling System (Windows Machine):** Acts as the interface to the entire X-Ray system. Any input has to be given through this system by the technician or radiologist.
- **Microcontroller (ATMega328p):** This microcontroller receives signals from the controlling Windows machine and then signals to the corresponding sub-systems or transceivers.

- **Serial to CAN Transceiver (MCP 2515):** This microchip module converts serial data from the microcontroller to CAN Data to be transmitted across the CAN-Bus line into the CAN Network.

The **System Collimator [Receiver –Rx]** section consists of three major components:

- **Serial to CAN Transceiver (MCP 2515):** This microchip module converts CAN data being transmitted over the CAN-Bus line into serial data for the internal microcontroller to interpret and issue commands through signals to various subsystems like the servo motors of the C-Arm.
- **Microcontroller (ATMega328p):** This microcontroller receives serial data from the CAN Transceiver and then it transmits out various commands to the other internal modules present inside an Artis System.
- **C-Arm Alignment Motors (Servo Motors):** These servo motors present in the collimator system for the proper alignment of the C-Arm with respect to the patient's body. This alignment decides the intensity of the exposure to X-Rays a patient has to encounter.

2.3 Standards

This section describes the standards used for the communication between:

- Artis system controller and the system collimator. – CAN Protocol
- CAN Data Logger and the Analyzing Linux/QNX. – Serial Comm. (RS232)

2.3.1 Between Artis System Controller & Artis System Collimator

The CAN Protocol followed by MCP2515 is CAN2.0b. This CAN Frame consists of seven fields: Start of Frame (SOF), Arbitration, Control, Data, Cyclic Redundancy Check (CRC), Acknowledge (ACK), End of Frame (EOF).

If SOF bit = 0 → Dominant.

If SOF bit = 1 → Recessive.

Table 2.1 CAN2.0b Message Frame

Field	Length (bits)	Description
Start of Frame (SOF)	1	Must be dominant
Identifier – Standard and Extended Formats	11	Unique identifier corresponds to Base ID in Extended Format
Identifier – Extended Format	29	Comprised of 11 bit Base ID and 18 bit Extended ID
Remote Transmission Request (RTR) – Standard and Extended Formats	1	Dominant in data frames; recessive in remote frames. In Standard Format, the 11 bit identifier is followed by the RTR bit.
Substitute Remote Request (SRR) – Extended Format	1	Must be recessive. SRR is transmitted in Extended Frames at the position of the RTR bit in Standard Frames. In arbitration between standard and extended frames, recessive SRR guarantees the standard message frame prevails.
IDE – Standard and Extended Frames	1	Must be recessive for Extended Format; dominant for Standard Format.
Reserved r0 – Standard Format	1	Must be dominant
Reserved r1, r0 – Extended Format	2	Must be recessive
Data Length Code (DLC)	4	Number of data bytes (0–8)
Data Field	0–8 bytes	Length determined by DLC field
Cyclic Redundancy Check (CRC)	15	
CRC Delimiter	1	Must be recessive
Acknowledge (ACK)	1	Transmitter sends recessive; receiver asserts dominant
ACK Delimiter	1	Must be recessive
End of Frame (EOF)	7	Must be recessive

Source: Michigan EECS

The mentioned fields are all the significant fields present in a CAN2.0b Message Frame. Some of the most frequently utilized fields are as follows:

1. **Start of Frame (SOF):** This denotes the start of a message and it synchronizes the nodes on the bus after being in its idle state.
2. **Identifier:** This denotes the priority of the message being transmitted (11-bit Identifier). The higher the binary value the lower its priority.
3. **Remote Transmission Request (RTR):** The bit denotes the acceptance/rejection of message based on its state. Dominant state denotes acceptance and recessive state denotes rejection of message.
4. **Control:** The control field denotes the length of the Data being transmitted.
5. **Data:** This denotes the actual 8-bit of data that holds vital information or signals being transmitted from one system to another.
6. **Cyclic Redundancy Check (CRC):** The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for error detection.

7. **Acknowledge (ACK):** After receiving a valid message, every node rewrites the acknowledge bit from a recessive bit to a dominant bit thereby indicating the acceptance of valid information. ACK consists of 2-bits: 1 acknowledgement bit and 1 delimiter bit.
8. **End of Frame (EOF):** This indicates the end of the CAN Message Frame and then it disables bit-stuffing.

2.3.2 Between CAN Data Logger and the Analyzing Linux/QNX

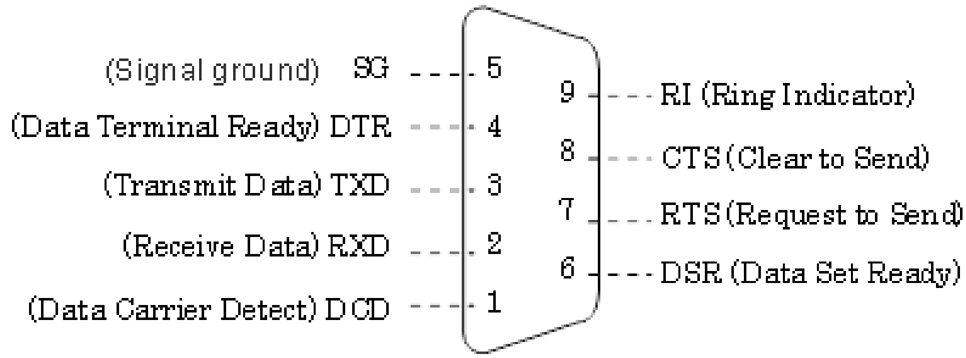
Serial Communication: The process of transmitting data bitstream sequentially over a bus is termed as serial communication.

Modes of Data Transfer in Serial Communication:

- **Asynchronous Data Transfer:** The data transfer mode which is not synchronized by a clock pulse.
- **Synchronous Data Transfer:** The data transfer mode in which the bits of data are synchronized by a clock pulse.

The basic characteristics of Serial Communication:

- **Baud Rate:** The speed with which the data is being transferred from the transceiver to the receiver in the form of bits per second. Few of the standard baud rates are 1200,2400,4800,9600,57600 etc.
- **Framing:** This denotes the no. of bits to be transmitted by the host device (transmitter) to the client device (receiver).
- **Synchronization Bits:** The transmitter appends to the original data frame with synchronization bits (1-bit for Start and Stop Each). This facilitated the receiver to understand the start and the end of data transfer.
- **Parity:** The corruption of data maybe occurs sometimes because of interference of external noise. The only solution to get a stable output is by the implementation of parity checking or verification.



Source: https://www.stratusengineering.com/rs232-9-pin-pinout/

Figure 2.3.2.1 RS232 – DB9 Connector for Serial Communication

- RS232 – DB9 Connector Pins:
    1. **Data Carrier Detect (DCD)**: Data terminal detection pin sends a signal to the data set that is going to be transmitted.
    2. **Receive Data (RxD):** Initial signal is received by the data set through the Receive data line.
    3. **Transmit Data (TxD):** Data terminal gets a signal from the data thereby confirming the connection establishment and the data set.
    4. **Data Terminal Ready (DTR):** Digital high (+5V) is applied on this line to indicate DTR is ready for transmission of Data.
    5. **Signal Ground (SG):** A return for all the signals on the same interface. Necessary for serial communication to take place.
    6. **Data Set Ready (DSR):** Digital high (+5V) is applied on the DSR line which ensures proper serial communication between data set and data terminal.
    7. **Request to Send (RTS):** A positive voltage applied on this line denotes the uninterrupted transfer of dataset to the data terminal.
    8. **Clear to Send (CTS):** Clear to Send signal denotes that the data terminal recognizes that communications can be performed.
    9. **Ring Indicator (RI):** This signal is activated when a low frequency signal is detected by the at the modem end.

2.4 Technical Specifications

Components:

- **Controller System:** It is a Windows OS based machine that controls the entire movement, functionality of the X-Ray collimator and the complete system. This machine transmits the necessary signals to the various internal components.
- **Microcontroller (ATMega328p):**
  - High Performance, Low Power AVR® 8-Bit Microcontroller
  - Advanced RISC Architecture
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48P/88P/168P/328P)
  - Operating Voltage: 1.8 - 5.5V for ATmega328P
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode.
- **CAN Transceiver Module (MCP 2515):**
  - Implements CAN V2.0B at 1 Mb/s
  - Data Byte Filtering on the First Two Data Bytes (applies to standard data frames)
  - Three Transmit Buffers with Prioritization and Abort Features
  - One-Shot mode Ensures Message Transmission is Attempted Only One Time
  - Interrupt Output Pin with Selectable Enables
- **High Torque Servo Motors:** These motors are present inside the C-Arm for the alignment of the collimator and the source of the X-Ray tube for the maximum aligned position.
- **CAN to USB:** It is used to convert the CAN Data to USB Data over a serial communication line.
- **Linux/QNX Machine:** This is the analyzing system on which the bitstream is visualized to interpret comprehendible CAN Data.

2.5 Summary

Thus, the methodology, design approach, standards implemented and technical specifications of the project were discussed in this chapter. The implementation of the CAN Data logger algorithm is explained in the next chapter.

CHAPTER 3

ARTIS SYSTEM COMMUNICATION & CAN DATA LOGGER

3.1 Artis System Internal Communication

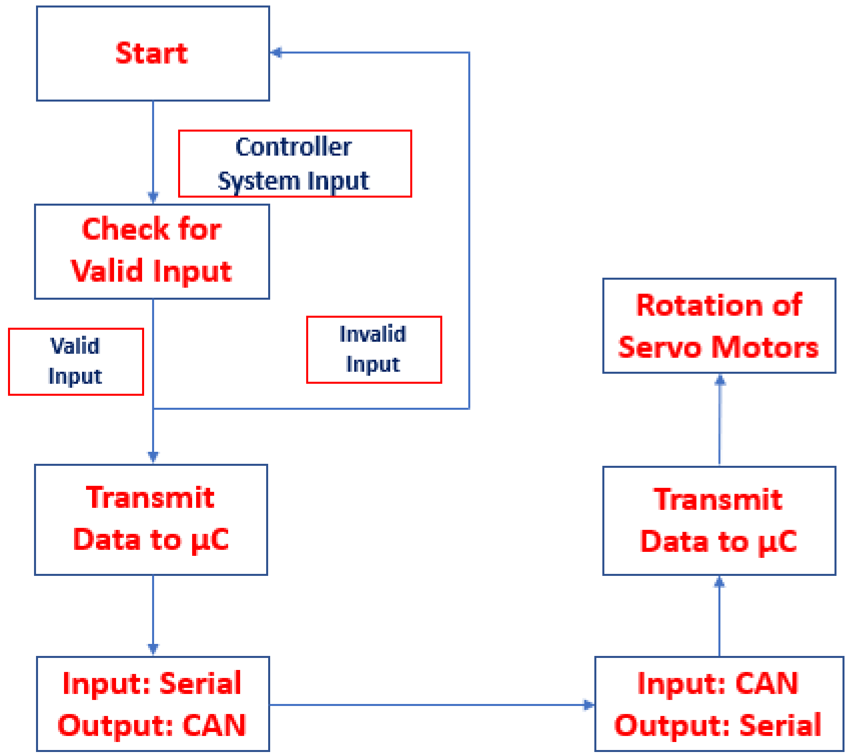


Figure 3.1.1 Artis System Internal Communication Algorithm

- The system is initially in the Standby Mode after being booted up.
- The internal algorithm then continuously checks for any serial input being provided into the system through the controlling Windows machine.
- If a valid input is provided into the system in correspondence to the internal functions and commands, a data signal is transmitted to the microcontroller.
- The microcontroller then again interprets the command received from the controller system and then transmits out serial data signals to the CAN Transceiver.
- The CAN Transceiver then converts the Serial Data to CAN Data with the help of MCP 2515 microchip on the transceiver module.

- This CAN Data being transmitted from the Artis System Controller goes to the Artis system collimator end.
- The CAN Data line at the collimator end is connected to a transceiver to convert the obtained CAN Data into Serial data.
- This serial data then is interpreted by the microcontroller and the necessary command signals is transmitted to the various peripherals connected in the Artis Collimator System.

3.1.1 Artis System Internal Communication – Code

Transmitter:

```
#include <SPI.h>
#include <mcp_can.h>

const int spiCSPin = 10;

MCP_CAN CAN(spiCSPin);

void setup()
{
    Serial.begin(115200);

    while (CAN_OK != CAN.begin(CAN_500KBPS))
    {
        Serial.println("CAN BUS init Failed");
        delay(100);
    }
    Serial.println("CAN BUS Shield Init OK!");
}

unsigned char carmf[1] = {1};
unsigned char carmb[1] = {2};

void loop()
{
  if(Serial.read()=='f'){
```

```
  Serial.println("C-Arm Forward - [Command Sent]");
  CAN.sendMsgBuf(0x43, 0, 1, carmf);
  }

  if(Serial.read()=='b'){
    Serial.println("C-Arm Backward - [Command Sent]");
    CAN.sendMsgBuf(0x43, 0, 1, carmb);
  }
}
```

Receiver:

```
  #include <SPI.h>
  #include "mcp_can.h"
  #include <Servo.h>

  Servo myservo;
  int pos = 0;

  unsigned char len = 0;
  unsigned char buf[2];


  const int SPI_CS_PIN = 10;

  MCP_CAN CAN(SPI_CS_PIN);

  void setup()
  {
      myservo.attach(7);
      Serial.begin(115200);

      while (CAN_OK != CAN.begin(CAN_500KBPS))
      {
          Serial.println("CAN BUS Shield init fail");
          Serial.println("Init CAN BUS Shield again");
```

```
        delay(100);
    }
    Serial.println("CAN BUS Shield init ok!");
}


void loop()
{

    if(CAN_MSGAVAIL == CAN.checkReceive())
    {
        CAN.readMsgBuf(&len, buf);
        unsigned long canId = CAN.getCanId();

       for(int i=0; i<=len; i++){

        if(buf[i]== 1){
        Serial.println("[Command Received]:C-Arm Rotated
FORWARD");
        forw();
        break;
        }

        else if(buf[i]== 2){
        Serial.println("[Command Received]: C-Arm Rotated
BACKWARD");
        back();
        break;
        }
       }
      }
     }

void forw(){
for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos);
```

```
     delay(15);
}
 }

void back(){
{
for (pos = 180; pos >= 0; pos -= 1)
     myservo.write(pos);
     delay(15);
}
  }
```

3.2 CAN Data Logger

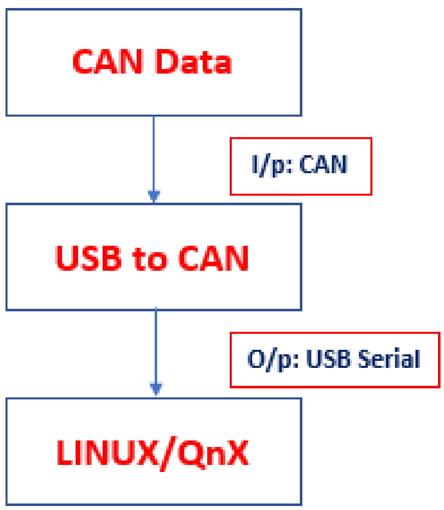


Figure 3.2.1 CAN Data Logger Algorithm

- The CAN Data being pulled out from the CAN Bus line is fed into a USB to CAN Converter.
- This Data is now available in USB Data format that is transmitted serially to the next visualization platform.
- This visualization of data is possible because the data received is in the form of USB Serial Data.

In accordance to Non-Disclosure Agreement and Siemens Compliance Policies any crucial company data (including literature, images, code) cannot be shared for thesis purposes.

CHAPTER 4
COST ANALYSIS

4.1 List of components and their cost

The costs of the various components used in this project are given below in Table 4.1.

Table 4.1 List of components and their costs

COMPONENT	COST
Atmega328 (2 no.s)	180x2 = Rs.360
MCP 2515 Module (2 no.s)	335x2 = Rs.670
Servo Motors (1 no.)	Rs.325
Jumper Wires	Rs.178
DB9 (RS232) to USB Cable	Rs.285
USB to CAN Converter	$ 29 (Rs.2005/-)
TOTAL	Rs. 3823/-

4.2 Cost Effectiveness

For the making of the final solution, a replica system of the actual Artis system was created on which the solution is initially implemented. And hence minimum hardware peripherals to make a proof of concept. So, for the final solution the necessary hardware would be the DB9 (RS232) to USB Cable and the CAN to USB Converter.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Artis System Replica

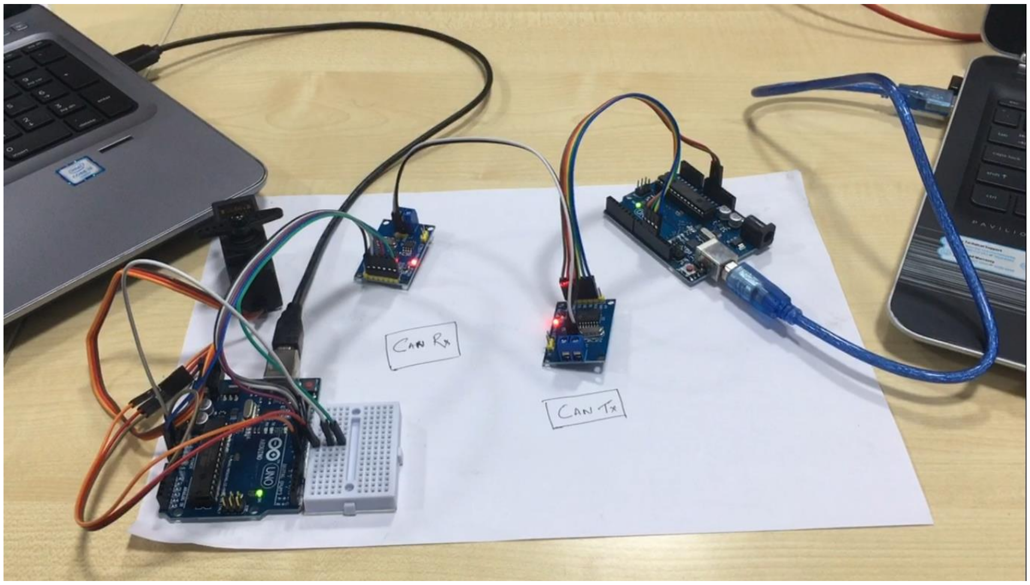


Figure 5.1.1 Peripheral Setup of Artis System Replica

This setup consists of the replica of all the hardware peripherals present inside a the Artis System Controller and the Artis System Collimator.

The replica setup into two sections:

- **Artis System Controller (Tx):**
  Input from System (Windows Machine) → Microcontroller data interpretation → Serial Data Transmit → CAN Module Transceiver → CAN Bus Line to Artis Controller.
- **Artis System Collimator (Rx):**
  CAN Data Input from Tx → CAN Transceiver → Serial Data → Microcontroller → Servo Motors and Other Peripherals.

5.2 CAN Communication

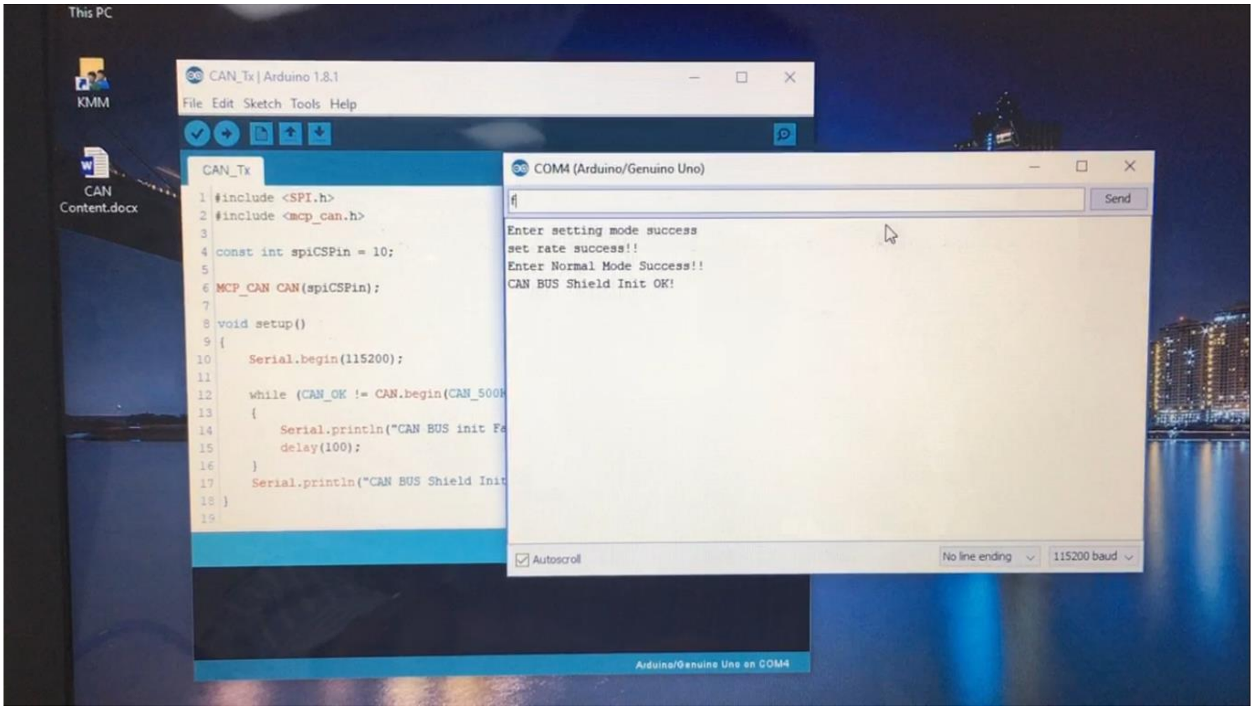


Figure 5.2.1 Successful Connection Initialization of CAN

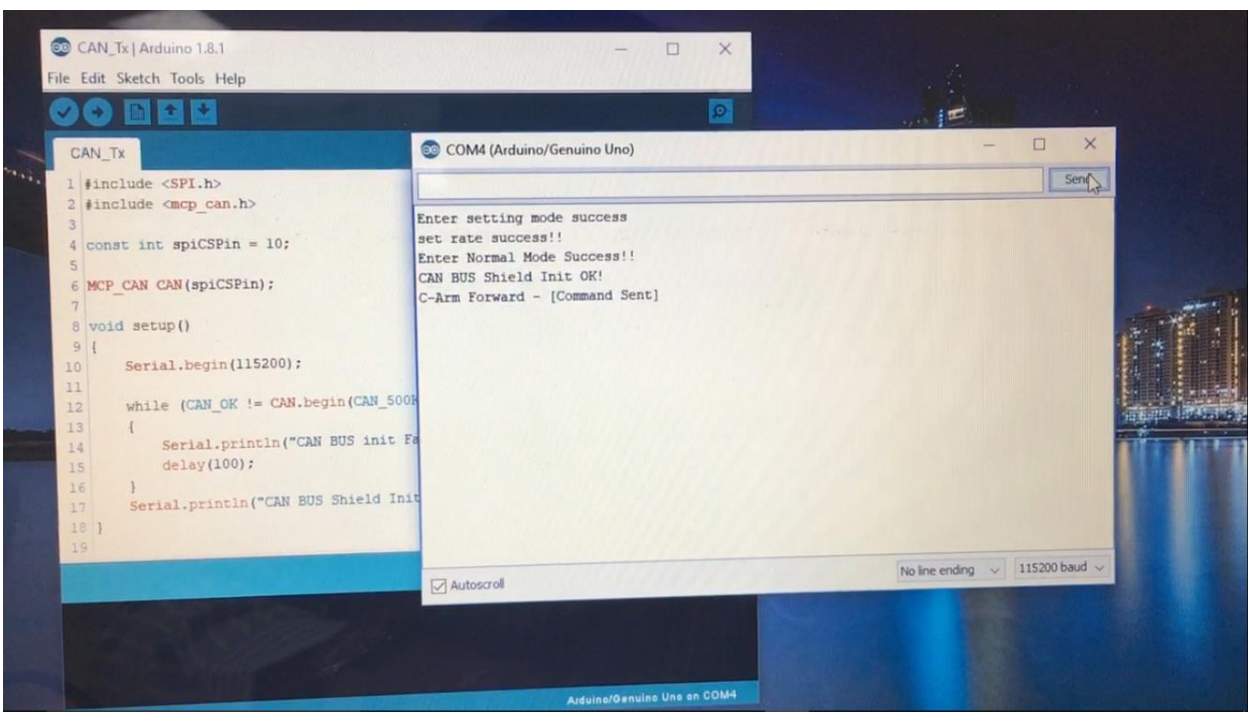


Figure 5.2.2 Serial Input Transmission from Controller to Collimator

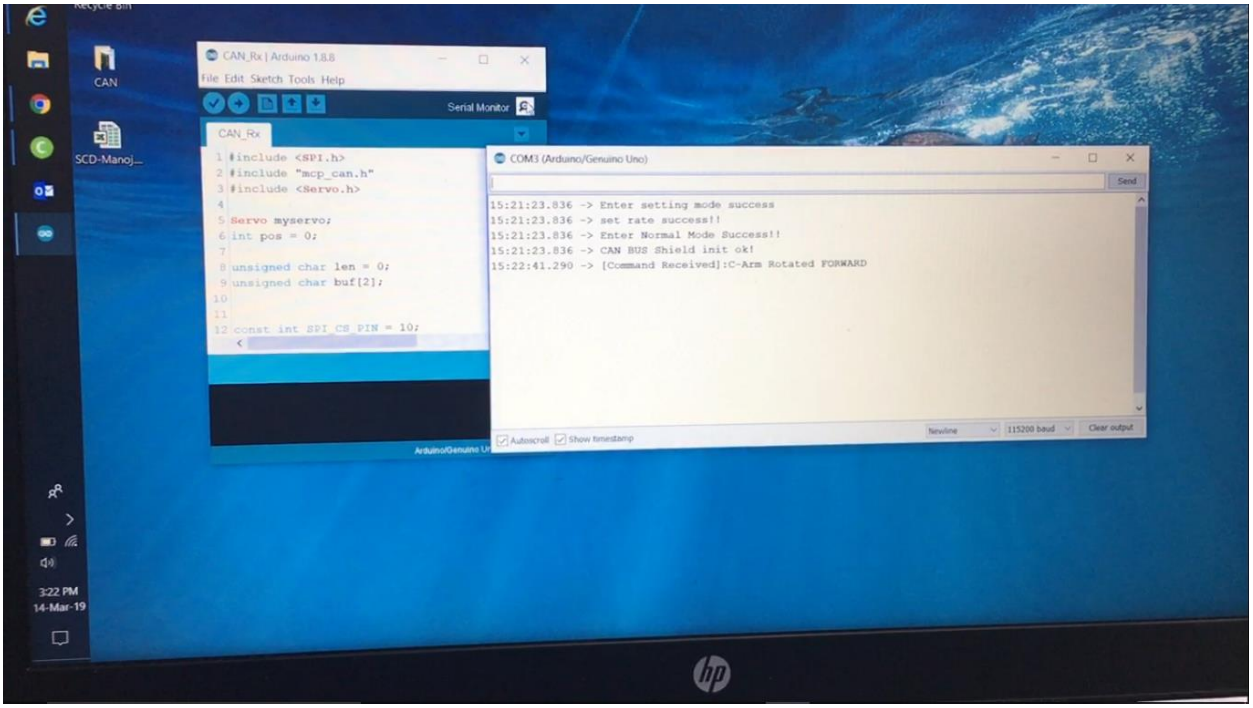


Figure 5.2.3 Successful Reception of Serial Data at Collimator

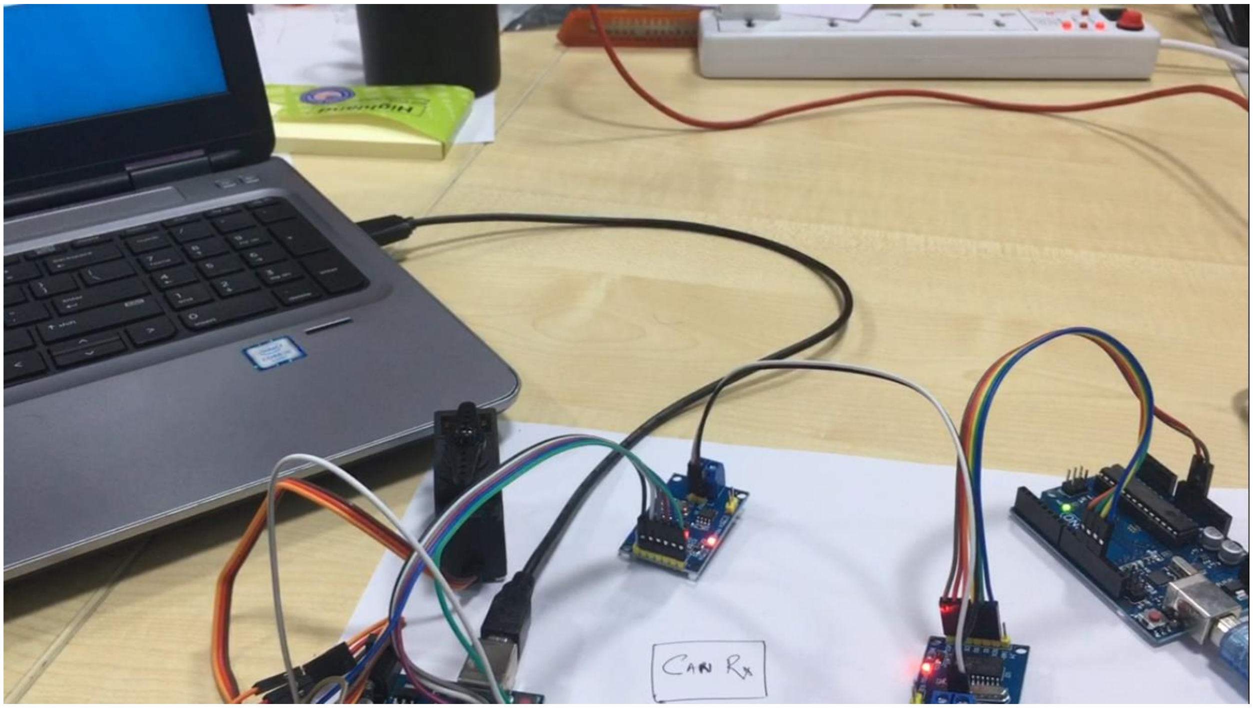


Figure 5.2.4 Rotation of Servo Motor at the Artis Collimator Side

In accordance to Non-Disclosure Agreement and Siemens Compliance Policies any crucial company data (including literature, images, code) cannot be shared for thesis purposes.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis on CAN Data Logger and also mentions the future work and scope of this project. This also mentions the implementation of the further stages of this addition of the existing system.

6.1 Conclusion

In this project, a method was devised to log all the CAN Data from Artis system controller and the Artis system collimator. This data being sniffed out is then fed into a visualization system running either Linux or QNX operating system.

This solution eliminates the need for sophisticated hardware additions to the existing Artis system to pull-out the CAN Data. The CAN to USB Converter handles the lossless data packet reception thereby not loosing out on crucial data.

6.2 Future work

The future work that can be implemented for this project is integrate this hardware addition at the production stage itself. This shall eventually eliminate the need for a service technician to visit the client site for any minor issues related to CAN.

CHAPTER 7

BIBLIOGRAPHY

7.1 REFERENCES

1. https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf

2. https://www.engineersgarage.com/articles/what-is-rs232

3. https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications

4. https://plugable.com/2011/07/05/installing-the-plugable-usb-to-rs-232-db9-serial-adapter-on-windows-7/

5. https://store.protofusion.org/product/canable/

7.2 LIST OF PUBLICATIONS

INTERNATIONAL JOURNALS

1. An Embedded CAN-BUS Communication Module for Measurement and Control System.
   Xiaoming Li, Mingxiong Li
   Mechatronics Institute, Faculty of Mechanical Engineering and Automation,
   Zhejiang Sci-Tech University,
   Hangzhou, China (310018)

2. Embedded On-Board Diagnostics System Using CAN Protocol.
   Pallavi R. Burje, Kailash J. Karande, Amol B. Jagadale
   International Conference on Communication and Signal Processing, April 3-5, 2014, India

3. Supporting Security Protocols on CAN-Based Networks.
   Gedare Bloom∗, Gianluca Cena†, Ivan Cibrario Bertolotti†, Tingting Hu‡, Adriano Valenzano† ∗Howard University, 2300 6th St NW, Washington, DC 20059, USA