LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY

School of Energy Systems

Degree Program of Mechanical Engineering

Amirali Abdolzadehgan

DEVELOPMENT OF ELECTRONIC SYSTEM FOR HORSE-RIDER INTERACTION IN A SIMULATOR

Examiners: Professor Heikki Handroos
Postdoctoral researcher Ming Li

ABSTRACT

Lappeenranta-Lahti University of Technology

Faculty of Technology

Master’s degree program in Mechanical Engineering

Amirali Abdolzadehgan

Development of Electronic System for Horse-Rider Interaction in a Simulator

Master's Thesis 2020

62 pages, 26 figures, 3 tables

Examiners: Professor Heikki Handroos
Postdoctoral researcher Ming Li

Keywords: Automation, Control, IoT, Force measurement, Simulation, MQTT

This master´s thesis is about investigating a wireless interface of the sensor system for transferring the data to the main controller to provide smooth and reliable monitoring and control for the simulator. This horse consists of various sensors which transfer important physical parameters to the main controller of the system. In this case Beckhoff PLC model: CX2030 is the main controller of the simulator. For the signal transferring, wireless method is selected to make this package more convenient and let riders to act on the horse more freely. By considering four degree of freedom, this design makes simulation closer to a real horse. This project aims to support a horse-rider trainee, to learn how to treat with a horse and learn how to ride it in different types of riding patterns such as walking, trotting and cantering. A real horse can run and practice for just one hour per day whereas a trainee can practice riding with this package all day long. In this field, there are not so many research and innovations available. This simulator has been designed by some companies, so far in this investigation I aim to develop wireless transmission interface to it.

The system which is developed consists of two main parts. First part includes Force sensor and inclinometer connected to the Arduino MKR wifi 1010. Calf angle and calf force from rider is

measured in this part. Second part includes Beckhoff PLC which is connected to the wifi router via LAN network. Sensor's signal from Arduino is transmitted to the TwinCAT software by means of MQTT protocol over wireless communication. The results of this thesis show that the communication can be conducted by using this architecture, and the long-time experiment shows that the communication is reliable without disconnection or missing data.

ACKNOWLEDGEMENTS

This thesis was carried out at Lappeenranta-Lahti University of Technology in the Laboratory of Intelligent Machines.

I would like to express the deepest appreciation to my supervisor, professor Heikki Handroos for his valuable supervision and support during my research. I am also grateful for the assistance and guidance provided by postdoctoral researcher Ming Li, for his special assistance and worthy comments during the research work.

I would also like to thank Juha Koivisto for his assistance.

Finally, I would like to thank my parents for their sincere support, love and encouragement.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASIC	Application Specific Integrated Circuit
CSB	Chip Select
FIFO	First In First Out
FS	Full Scale
FSR	Force Sensitive Resistor
IDE	Integrated Development Environment
LED	Light-Emitting Diode
LPF	Low Pass Filter
LSB	Least Significant Bit
MCU	Microcontroller
MISO	Master In Slave Out
MOSI	Master Out Slave In
MQTT	Message Queuing Telemetry Transport
POU	Program Organization Unit
QoS	Quality of Service
SCK	Serial Clock
SPI	Serial Peripheral Interface
SSL	Secure Sockets Layer
STO	Self-test Output
TLS	Transport Layer Security
UTF-8	8-bit Unicode Transformation Format

LIST OF SYMBOLS

I	Electrical Current, Ampere (A)
P	Electrical Power, Watt
R	Resistance, Ohm (Ω)
R_M	Pull down resistor, Ohm (Ω)
V	Voltage, Volt (V)

1 Introduction

This thesis will introduce and make the reader familiar with wireless protocols and solutions used in automation. Some exclusions had to be made, to keep the topic from expanding too much.

1.1. Scope of work

The research conducted in this thesis will only address the instrumentation and sensor interface of the horse-back riding simulator. Due to the wide range of research area involved in this simulator, other dynamic and behavior design was decided to be done by other experts in mechanical engineering department of LUT university.

As for the main controller systems, only supported protocols and approaches will be addressed, and focuses more on the wireless solution itself. The programming of the logic and network in Beckhoff PLC and its different monitoring and control systems have more than enough variables to write up this thesis in the time window given. However, this research can be expanded to entire control system and different point of views in further research if necessary.

The scope of work in this research is divided to bellow tasks:

• Transmitting calf aids to the simulator controller which is composed of:

- pressure sensor to measure the calf pressure against the horse body
- angular sensor to measure the calf and stirrup angle
- installation and integration to the simulator controller

• Reins tension measurement and transmission of this signal to simulator controller

• Integration of the sensor mentioned into the controller via wireless technology

1.2. Research problems

The topic of the project is quite broad and covers many different sensors and control strategy such as Bluetooth, WLAN, etc. Considering the scope of the research, the topic is still quite challenging and has many different branches. Therefore, the research problem is quite nonrestrictive and has quite a big range of possible answers and solutions:

- There are plenty of wireless concepts and solutions which have different purposes and uses. When to use which kind of wireless systems?
- How to integrate the sensor system with main controller? Since each system manufacturer has their own software and framework, therefore connecting these different networks needs more precise research and attention.
- Installation and operation principle of the sensors are another challenging part of this research. Where there are many acts will take place in riding a horse and special preventive solution should be taken into account in case of maintenance and reliability of the sensors and their connections.

These problems themselves do not have a definitive solution, but this research will suggest different perspectives and solutions for some cases in wireless communication between two different systems.

1.3. Objectives

The goal of this thesis is to solve the practical problems and presenting a feasible solution according to the requirement of the project. Some of the equipment's are widely used and others quite rare. The need of wireless communication will also be explained to make the reader understand exactly why this method is selected and what kinds of advantages it will bring to the users. connection lost and data loose are a risk that can be managed with this method as well.

Research questions in this report are the following:

- Why wireless approach is used for this project?

- What is the most efficient protocol for this project?
- How these sensors can transmit their signals to the simulator's controller?
- Which kind of modifications and future studies are needed in this area?

1.4. Motivation

Horse riding is famous sport and has a long history. It has some effects classified as psychological effect in terms of relaxing gained by riding and in terms of movement against gravity or keeping the balance on the saddle. Focus in horse riding is on correcting the rider's flexibility and balance feeling. There are also other criteria that are important in this sport, such as entertainment and physical therapy. Horse- riding assisted therapies is very well known in today's medical sciences. The main therapies include hippo therapy and therapeutic. That therapies use for physical remedy. On the other hand, horse riding does not have popularity in huge urban areas, because of cost, learning difficulty and injury danger. For solving these issues, several robots have been introduced for simulating this riding. (Shinomiya et al., 2003)

The horseback riding simulator works like a machine which simulates the motion of riding in an emulation of the basic movements of a real horse (Chen et al., 2002), (Eskola and Handroos, 2013). common simulator is expanded to act such as a real horse, therefore similar experience effects to the riding of a user in a real world. The result of this interaction is also considered. Basically, providing a continuously interest to riders is not so easy especially when they are using the simulator instead of real horse. for dominating this issue, one possible approach could be developing a more user-friendly hardware and software.

For instance, in Korea government is tending to improve the people's life quality via developing the horse industry and promoting people in this field. One of the famous companies in UK is Racewood Ltd. Which is recognizes as the pioneer company in this industry. They are actively working and invent new features and methods to improve horseback riding simulators. In recent years, horseback studies carried out (MacKinnon et al., 1995).

Another considerable benefit of horseback riding therapy applies to elderly people who suffer to low-back pain and balance difficulties (Nakajima et al., 1999). Some scientists argue that horseback riding has many effects such as increase of the muscle strength, and this proved by most of clinical experiment. Effect of the horseback riding into each human body is individual, therefore an appropriate type of horse motion should individually be selected.

The topic of horse-riding simulation becomes more widespread and critical these days due to technological advances in these decades. Availability of robot's platform and new simulation software are the main reason for this improvement. Basically, developers must write diverse logic program to meet the need of each simulator configuration. The important role that should consider is integration and communication between all parts and taking an appropriate result at the end. One particular area for employing a horse-riding simulator is the riding schools. This can improve horse's and rider's welfare and allow the riders to practice more before trying a real horse. Some popular sensors included into this project is pressure sensors for calf pressure sensing and load cell applicable for reins tension detection. Another area that will consider in this thesis is integration of these control signals into PLC based control system via wireless communication. In this thesis I will focus on sensors, control and transferring the signal to the controller for decision making and appropriate actions.

2 Literature review

This part will describe how the research was conducted and which databases were used as a reference. Complete reference list is provided at the end of the report. In this section the methods of the literature study will be explained and defined. In the process of preparing the thesis, many databases were used, such as Scopus, science direct, IEEE, etc.

The data analysis tools in this database over the past 13 years were also used: the graph shows a continuous increase in the number of publications on this topics, which makes it possible to make sure that the importance of the topic of IoT has increased over the past decades. Results are shown in figure 2.1.

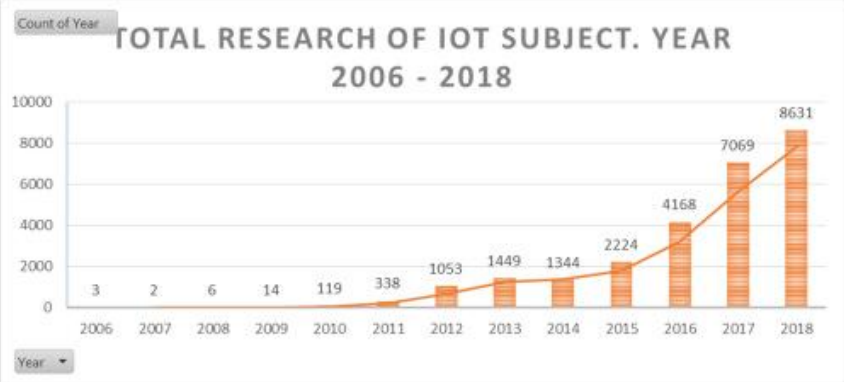

Figure 2.1. Total research about "IoT" subject. From 2006 to 2018 (Dachyar et al, 2019).

Gathering the source information for general knowledge of horse-back riding simulation and different control systems was conducted using LUT Finna search engine.

2.1 Design

Various simulators and robots have been produced to simulate the riding, (Shinomiya et al.,2003) developed a device applicable for horse-back riding therapeutic with 6-DOF parallel mechanism. This simulator has the trainee persuade muscular discharge efficiently for his/her body and prepare the sense of horse-riding. In long-term usage of that simulator outcomes in an increase of muscular strength of individuals and betterment in the insulin sensitivity of patients suffer with type 2 diabetes. The features of the developed simulator is a common movement which is sufficient to experience the sense of horse riding and able to muscular discharge- In their research they express that elderlies suffer to back and stomach muscle weakness and sometimes standing and walking conditions tend to be garbled, however by increasing the load on the lower back it will lead in decreasing back pain. So, they started to think about specific training that can increases the power of these muscles. statistics shows that over 6 million people who suffer with type 2 diabetes only in Japan. and it is famous that type 2 caused insulin resistance and reduced glucose absorb to skeletal muscle. By checking the result of physical activities training in this groups of patients, significant improvement achieved in glucose uptake

and insulin sensitivity in their whole bodies (Sato et al., 1991). But since the elderlies tend to have less activity that equipment becomes more important to assist them continue their exercises and physical trainings. These were the main aims of their projects and research. In their design horse movement represent as a 6-axes parallel system. In that mechanism motion capturing technology used as the main idea. Saddle movement captured as three-dimension sequence data. That mechanism represents the horse movement and its effects on the body in terms of movement speed and pattern. They decided to reduce the scale of their mechanism in order to become more cost efficient and labor acceptable. for overcoming to this goal, they carried out two separate measurement experiments to produce a simplified their design and include riding feel with more attraction and muscular discharge growth. In that research they tested 11 riders without any experiences in riding. The evaluation technique was satisfaction measurement in riders randomly. After some test and experiments, it was conducted to verify the result of the horseback riding equipment. It has effects on the strengthen of abdominal and back muscles as well as empowering the knee joint another goal regard to reducing in insulin resistance in the body of diabetic patients (Shinomiya et al., 2003).

Also, (Amirat et al., 1996) invented a hydraulic operated simulator. It was based on parallel robot. That mechanism executed from a classic Stewart platform. There were some hydraulic jacks, links and actuators include in the research. The main application of that robot is an equestrian simulator and the related control architecture have been implemented from signals which have been achieved from the real horse. That parallel robot includes of some moving parts such as end effector and some fixed parts such as independent kinematic chain. The reasons that they used parallel robots instead of series robot were that, 1- high rigidly because of their mechanism. 2- a homogenous dispensation of inertia and 3- non-accumulation of joint position error. Also, they expressed the disadvantage of parallel robots as its workspace limit. (Amiratet al., 1996) used Stewart type platform in his design which has weak inertia of the parts equipped with hydraulic jacks and actuators. The result was high dynamic performances due to high loads moving ability with high acceleration. Equestrian gait is completely like that, as it does not need large displacement, but it needs high acceleration. The mechanism consists of a fix base, some mobile parts and six legs connecting to each other. Each leg consists of two spherical passive

joints and one prismatic active joint. Their research includes inverse geometrical model and inverse kinematic model. their modeling and calculation required real-time control of the mechanism and that model have been computed from a time estimation point of view. In case of control system, it is actuated via six differential hydraulic jacks. Each jack is actuated with an electro-hydraulic servo-valve which is energized with an analog servo-amplifier that involved a PID controller as well. The hydraulic supply pressure is 140 bar with 100 l/min flow rate. The aim of the research is to simulate the acceleration produced by the rider during the experiment. First from data acquisition system various gaits data gathered. Linear and angular velocity is measured respect to the main frame.

(Lee et al., 2018) claimed that 6-DOF with parallel mechanism simulator have a small working space and the orientation capability of it would be very poor. but the actual simulator has smaller displacement than a real horse. For making this size larger and improving the orientation capability, size of the mechanism should become bigger. However, making the mechanism bigger may lead to occurring interferences between the components or maybe rider's body. For preventing from this issue, they introduce HKM concept, which consists of serial kinematic mechanism. This mechanism splits the manipulation into position and orientation. By this concept they were able to use serial kinematic for orientation and parallel mechanism for positioning of the whole structure. By mounting a moving plate on the bottom of the design and rising the center of mass and rotation, they were reduced the size of the platform compared to integral structure and prevent accidentally interference between rider body and the simulator.

Meantime, (Chen et al., 2002) introduced a horse-back riding simulator equipped by 6-DOFStewart principles with extra computer for having the virtual reality besides the simulator. Moreover, (Eskola and Handroos, 2013) invented a simulator that discovered three important gaits: walking, trotting and galloping according to 6-DOF hydraulic concept of Gough-Stewart platform.

2.2 Motion simulation

Moving platform of the simulators are designed based on the parallel robotics concept. In this method four legs considered between the base plate and moving plate and each of them include linear actuator and universal joints and prismatic joints in the structure. Simulator control conducted by multi-actuator controller equipped with I/O interface modules. The applied network is EtherCAT, due to its high data transfer speed (100 Mbps) the performance of the simulator increased significantly. The platform also equipped with two beat, bridle and preparative sensors. Sensor's signals transferred to master motion controller by using EtherCAT network. additionally, they insert rotation module on the moving plate to prepare pitch motion as well as rising the center of structure by adjusting the height of moving plate low. (Lee et al., 2018). In other researches such as (Lee et al., 2013), scientist started to investigate the analysis of the athlete's 3D motion while horse riding. They believe that horse riding can provide some psychological and social satisfaction by providing a harmony between horse and human. They performed 3d motion analysis equipped with visualization for their simulator. The also asked national horse rider experts to wear sensor-based clothes, to capture the motion of their bodies. They attempted to perform 3D motion analysis according to angle of knee, elbow and backbone. Three types of motion capture presented as optical, magnetic and mechanical. mechanical type attached to the human's body for capturing the motion, however magnetic type sense the motion by changes detection in the magnetic field. The drawback of this sensor is losing the magnetic capability after a while make the measurement unreliable. (Lee et al., 2013) decided to apply optical sensors for the motion capture. By using this type of sensors, they were able to extract the required information of the rider's position as well. Beside this advantage, optical sensors bring flexibility and smooth capture of motion data to the study. Motion data format repartitioned in to undefined and defined skeleton categories. They used defined skeleton data for 3D motion presentation. In their experiment phase since they want to simplify the experiment, they just consider walking and trotting gaits. (Li et al., 2011) conducted a study about riding postures in different gaits with different speeds with the aim of finding a simple method for evaluation of jockey's condition control ability. They concern about the low amount of scientific research in this field may affect the behavior and relationship between horse and

human. Performance of each riders was recorded by means of video and image recognition software. They measured vertical displacement of body and vibration characteristics of simulators hip. (Chen et al.,2002) introduced a biofeedback controller for horse-riding simulator. They considered the simulator as an empirical therapy for mental and physical disease. They applied a portable electro myometer to construct the biofeedback system architecture. In this control system, adjusting the displacement of the simulator for achieving exercise intention is proposed. That system consists of Stewart platform with a horse body mounted on it. Whole process was controlled with reins and force on the flanks and comparison with data from real horse. For the saddle displacement, analyzing and measuring of the movement conducted via high-speed photography. They applied six high-speed cameras for capturing the data of horse gaits. The aim of using electro myometer was to generate a biofeedback controller to optimize the simulator movement. (Koenig et al., 1993) presented a simulation method for generating and controlling the lateral gaits in horse simulators. They introduced a kinematic model for real time controlling and generating the gaits. In their method riders input entered to the neural network and the output of the process will be the input of the gait. In this method, gait is generated by means of finite state controller. Kinematic joint data is the input of finite state controller. The kinematic description in their design has 12 degree of freedom in body and 9 DOF in each leg. The also considered 6 DOF for position and orientation additionally. Use of finite state architecture allows them to demonstrate the motion of particular DOF of a horse. They also defined five critical cases and five stop states for motion. For each state, some features are described, such as angular velocity, transition criteria, trigger joints and a datum. Proposed simulation loop consists of four stages: Evaluate, render, upload and transform. by using this method, they execute required signal in control system. control system contains a transition, when all conditions are satisfied it will go to the next stage. This transition leads the simulator to change from current gait to desired one. eventually, their algorithm mapped text-style commands of the rider to numerical values for compile.

2.3 Previous research results

(Lee et al.,2018) analyzed and tested the horse gait for a long time by using different types of sensors. His aim was to improve the result. Their method was to classified horse gait into walking, cantering, trotting and galloping and measuring the leg cycle time for each pattern. In their research, horse motion data for each gait was captured, and they match this data with the horse skeleton. They also simulate the saddle and foot contact of the rider's in each type of gait. The platform provided 4 DOF which developed the workspace and slightly improves the orientation ability beside reducing the weight of structure. Their saddle design generated signal based on contact time between foot and saddle. (Lee et al., 2013) in 3D motion analysis of the horse riders body shows that the graph of backbone movement for national team horse rider and a beginner rider. They visualized the flexion position and angles in trotting gait. By these experiments they provided a motion database which is based on 3D skeleton motion analysis of the professional riders. (Li et al., 2011) investigated vertical displacement and phase characteristics of the jockey riders. The measured that the vertical movement of the riders are around 23.1 cm. But in phase characteristics experiment, they indicated that, there was a close relation between the skill of riding to adapt the rider with the harmonic motion of the horse. This adaption will lead to energy and cost saving for both human and horse. (Byeon et al., 2013) searched about the development of international and domestic horse-riding simulators. They studied several domestic and international resources of the simulators with sufficient detail about their design features and characteristics. And at last they conducted that, due to lack of some critical parameters such as reality of simulator, low DOF and lack of coaching system, users are not to tend to use their simulators. Also, they proposed a coaching system by constructing a database from national team's members by capturing their motion and make the distance of the real horse and simulator closer. (Chen et al.,2002) by using an electro myometer and generating a biofeedback represent that the relation between movement gait and electro myometer output is unvaried in a short period of time. Measurement time of their experiment was 180 seconds, in this period, they extract 9 movement modes. Their results show that developed biofeedback control solution can adjust the movement modes, additionally due to close relation between system output and displacement modes of the horse simulator, this data

can kept just for a short time but they were not specified the length of that period. (Koenig et al., 1993) introduced a finite state architecture for the control systems to present the motion of simulator by using transition and flowchart. By use of neural network, rider's inputs will provide simulator's output. By training the simulator, which is equipped with neural networks, simulator is able to learn and map the signals between rider and function codes. They visualized three gaits of a horse in animation and they believe that their experiment was very realistic and close to the real horse motions.

3 Methods

3.1 Presentation of components

3.1.1 Micro Controller

The idea of IoT is one of the well-known solutions in most of the automation projects. Using a small microcontroller with wireless communication possibility is the main target for selecting "ARDUINO MKR WIFI 1010". This board has one of the easiest entry points in IoT projects. Wireless capability of this microcontroller makes it very applicable and reliable. It can be connecting to all network routers and wireless networks in home or office, by a simple set up configuration. The powerful "Arm® Cortex®-M0 SAMD21" with 32-bit processor architecture beside low power consumption chipsets and also ability to power it via Lithium polymer battery, makes it as one of the best choices for this study. Figure 3.1 shows the microcontroller.

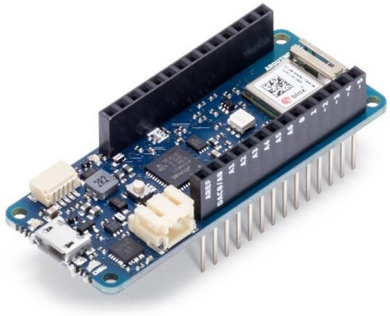

Figure 3.1. ARDUINO MKR WIFI 1010.

In addition of wireless capability, this microcontroller equipped with Bluetooth® connectivity. The board power supply is 5 VDC and power consumption of it is less than 20 mA. It supports all networks for data transmission, such as UART, SPI and I2C. Input/output power consumption is 7 mA. Figure 3.2 presents the pins connection configuration of the microcontroller.

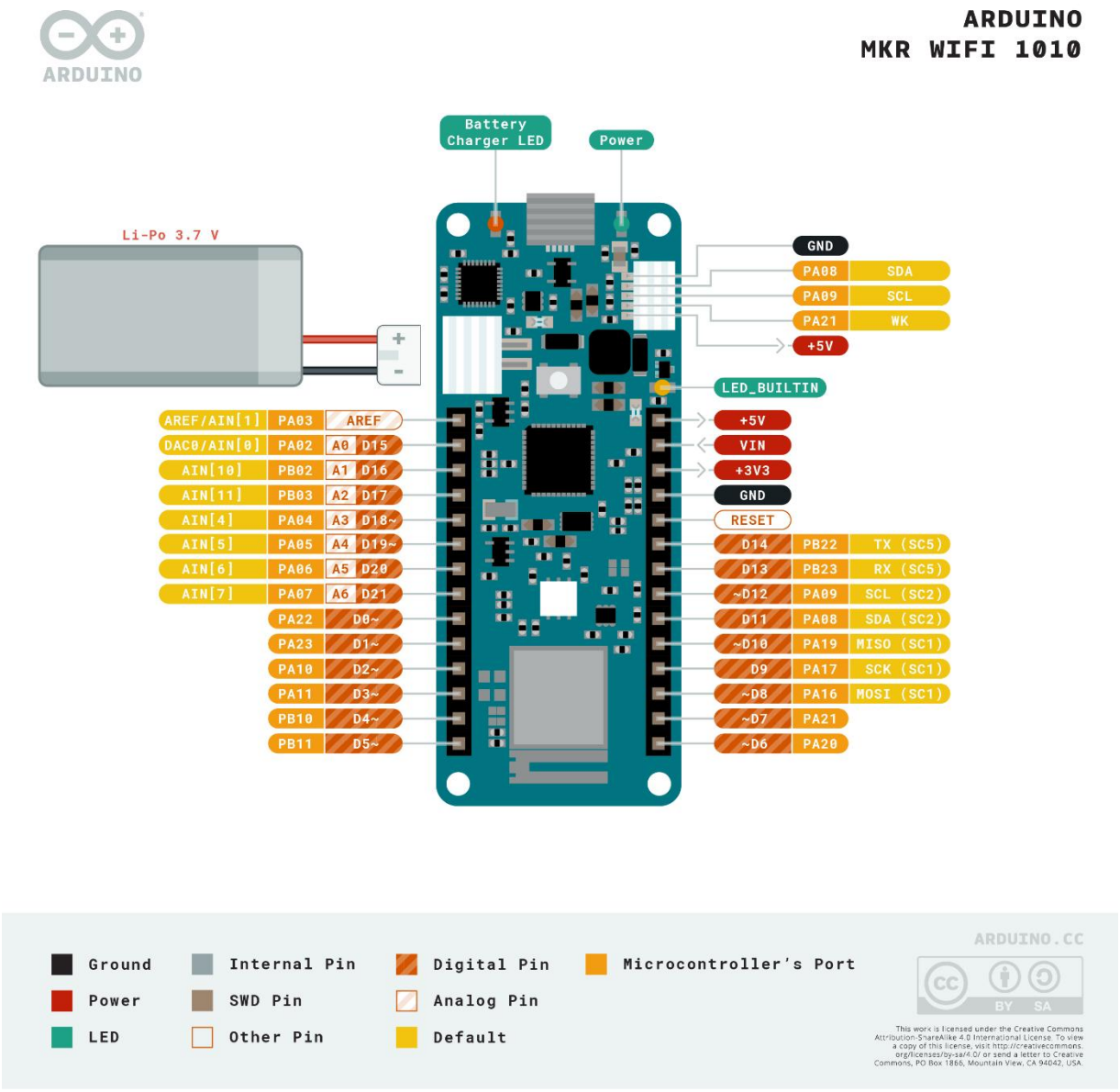

Figure 3.2. ARDUINO MKR WIFI 1010 Pins connection configuration.

3.1.2 Calf pressure

Horse-back riding simulators needed several physical parameters which are generated by the riders. These physical parameters are Force, pressure and acceleration which are mostly common in each simulator's platforms. One of the important parts in case of detection and measurement of the parameters is calf muscle. Rider generates command to the horse via his/her calf muscle. By applying force to the body of the horse and with its duration horse will understand what the aim of the riders is and how to behave. Although horses are so different in case of the reactions and sensitivity to their riders but in this research, I considered a unique type of behavior for simulator. Detecting the calf pressure against the horse body is the main aim for installing a force sensor in this part of the research. Most of professional horse riders believe that training the lower leg before any other part of the body is the key to good riding! Therefore, for having this good riding, the posture of the rider's calf is very important. This can bring stability, control and strength in horse riders. Riders lower leg should not be too far forward and not solid against the horse's body. Avoid pushing the heel to the horse body is the most important point of this part. This is a common issue that most riders have it. sometimes their lower leg swings forward or on the way far from horse body. This issue mostly happens when riders are in trot situation. The key muscle for control of the lower leg movement and make it stable is hamstrings. The first skill that each rider should learn is to sit solidly against the horse body and do not put any pressure on horses' body. Considering the good balance makes riders safer confident and ability to support their seat and body. Putting pressure on horse body is a communicative action between horse and human, This action could be a painful and annoying activity for horse if the riser is not aware about how to perform it or it can be a very joyful action for horse if the rider do it in a correct way. The heel should not fall too low or should not compress in high level than fingers.

Force Sensitive Resistance sensors (FSR) have a vast application in industries such as weighting, electronics, gaming devices and robotics. These sensors are super sensitive and reliable in most of measurements experiment. They are exclusively designed for force pressure sensing. In case of sensor structure, it consists of two membranes and one spacer adhesive between them. Membranes are conductive but separated with a very small air gap in between,

this situation happened when there is not any pressure applied to them. One of the membranes includes traces which is connected from sensors tail up to sensing area of the sensor are not touching each other. Another membrane which is covered by conductive ink will connect to another membrane when force applied to the sensor. The short-circuit between ink and two traces will make change to the resistance of the sensor. Applying force increase the resistance output of the sensor. The body type of the sensor constructed flexible and this feature helps to install it on non-flat surface without huge curved required. Figure 3.3 shows the applied force sensor in this study.

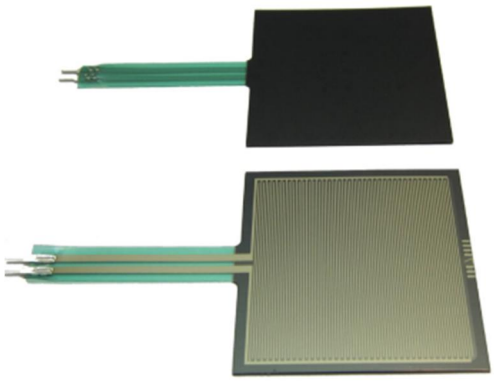

Figure 3.3. FSR 406 (Force Resistive Sensor) 38.1 x 38.1 x 0.54 mm.

This sensor can convert force of the calf muscles to the voltage. This voltage can be measured in Arduino microcontroller.

Normal size of the sensor is 43.7 x 43.7 mm (LxW) which is possible to be ordered in the size of 600 mm. The active sensing area of this sensor is 38.1 x 38.1 x 0.54 mm (LxWxH). by increasing the force, the output voltage will increase.

3.1.3 Angle sensor

Rotation degree of the stirrup is the next parameter that should be take into account. There is one single angle is needed. When the rider moves his/her foot forward the stirrup, this makes an

angle with respect to axis perpendicular to the horse body. This means “raise canter” command to the horse. The rider gives this command to the simulator controller when he/she wants the simulator to start the canter motion. Measurement of correct angle will improve the performance and balance of the simulator. This study conducts a solution for stirrup (ankle) angle measurement. To achieve this angle, an inclinometer sensor considered for installation on chaps. Selected sensor is “SCL3300-D01”. Figure 3.4 shows the angle sensor applied to this study.

Figure 3.4. SCL3300-D01 Murata angle sensor.

This sensor has measuring capability for three axis (X,Y,Z) That means that the angle of rotation in all three axis space can be measured by this sensor. The output interface of the sensor is based on SPI digital interface. This means that there are 4 pins for transmitting data between sensor and controller. This sensor has three selectable measurement modes for tilt measurement. The first mode is in range of 3000 LSB/g with 70 Hz LPF, the second mode is 6000 LSB/g with 40 Hz LPF and the last one is 12000 LSB/g with 10 Hz LPF. The angle conversion to degree is done according to equation 3.1.

$$Angle\ (°) = \frac{ANG_{\%}}{2^{14}} * 90 \qquad \text{(Eq. 3.1)}$$

In this equation, ANG_% is the angle output register included (ANG_X, ANG_Y, ANG_Z) in decimal format. Basically, of this sensor includes acceleration sensing element and ASIC. The acceleration sensing element transmit the signal to the AFE (saturated signal) and ADC convert the analog signal to digital by using a signal conditioner this digital output will transmit via SPI protocol to the user. Table 3.1 describes the pin configuration of this sensor.

Table 3-1 *Pin description of angle sensor*.

Pin Number	Name	Type	Description
1	AVSS	GND	Analog reference ground, connect externally to GND
2	A_EXTC	AOUT	External capacitor connection for analog core
3	RESERVED	-	Factory use only, connect externally to GND
4	VDD	SUPPLY	Analog supply voltage
5	CSB	DIN	Chip select of SPI interface, 3.3V logic compatible Schmitt-trigger input
6	MISO	DOUT	Data out of SPI interface
7	MOSI	DIN	Data In of SPI interface, 3.3V logic compatible Schmitt-trigger input
8	SCK	DIN	CLK signal of SPI interface
9	DVIO	SUPPLY	SPI interface supply voltage
10	D EXTC	AOUT	External capacitor connection for digital core
11	DVSS	GND	Digital reference ground, connect externally to GND. Must never be left floating when component is powered.
12	EMC_GND	EMC GND	EMC ground pin, connect externally to GND

3.1.4 Saddle pressure

Fitting the saddle on back of the horse is well recognized as one of the important factors for monitoring the rider's pressure which is applied to the horse. It also can help trainers to find the sitting problems of their trainees in the riding sessions. By respect to the horse's center of mass, there are various ways to define rider's sitting pattern. Saddle is always mounted on the same location. The rider can also steer the horse by varying the pressure distribution. When the rider leans forward, it means increase speed, while leaning backwards gives the horse a signal to

decrease the speed. One of the famous methods for measuring the pressure of the saddle is by using the pressure mats under the saddle. These mats consist of matrix based tactile sensors which works by principle of piezo resistance. Sensing cells covered all the mat's area and provide pressure analysis at each contact point. They can provide real time data about the pressure which is applied to them. In this study pressure mats did not considered as the pressure sensing system. Due to some restriction in the software of them, that is not possible to integrate to the main controller and they need their own software for reading the data. Moreover, the price of these pressure matts is very expensive. So, the proposed system for the saddle pressure measurement is considered from load cells. which are installed in center of the saddle and each corner of it. Load cells are specifically use for force transducing. They are converting the force such as tension, compression or torque into electrical signal. Figure 3.5 shows the load cell applied in this design.

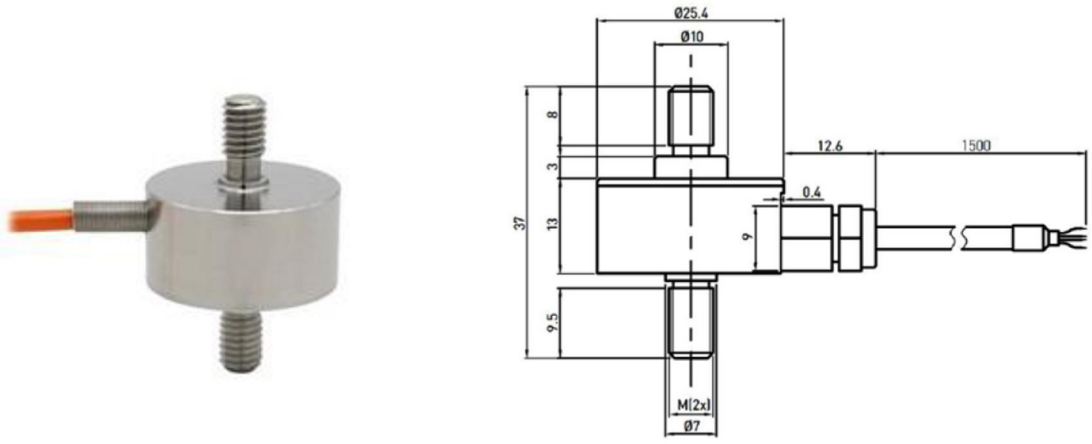

Figure 3.5. Compression load cell applied to the design.

By considering the capacity of 50KG for each load cell and considering 5 load cells distributed under the saddle area the measurement of force can be conducted. The input resistance of this sensor is 400 Ω±10, This means that by applying a 10 VDC to the input wire of this sensor according to the equation 3-2 the current can be calculated:

$$I = \frac{V}{R} = \frac{10}{400} = 0{,}025\ A \qquad \text{(Eq. 3.2)}$$

This equation express that the current consumption of each load cells is 0.025 A and all of the five applied load cells will consume 0.125 A. Due to its resistive type of load, It is possible to calculate the power consumption of the load cells as well. According to Joule's law and combination of it with Ohm's law, below equation present the power consumption:

$$P = I * V = I^2 * R = \frac{V^2}{R} = \frac{100}{400} = 4\ Watts \quad \text{(Eq. 3.4)}$$

Table 3-2 represent the technical parameters of the load cells:

Table 3-2 *Load cell's technical specification.*

Operating range	50 Kg	**Insulation resistance**	≥5000 MΩ
Rated output	1.5 VDC	**Operating temperature**	-30 ~ 70 ºC
Comprehensive error	±0.5% F.S	**Safe load limit**	150% F.S
Creep	±0.5% F.S	**Ultimate load limit**	200%F.S
Zero balance	±1% F.S	**Recommended excitation voltage**	10 ~ 12 VDC
Temperature effect on output	±0.05% F.S/10ºC	**Maximum excitation voltage**	15 VDC
Temperature effect on zero	±0.05% F.S/10ºC	**Protection class**	IP67
Input resistance	400±10Ω	**Output resistance**	350±10Ω
Connection detail	Red: Ex+; Black: Ex-; Green: Sig+; White: Sig-		

3.1.5 Reins tension

Reins are mostly used for directing the horse and it is considered as one of the horse aid facilities. However, it can be used for slowing down the horse and work as a harness for decreasing the speed of the horse. In the real horse riding, it can be a signal from rider to the animal to start to move. By having this background for reins application for the horse-riding simulator, the proposed reins tension sensor for measuring and transmitting the rider's signal has been carried

out. By investigating the physical principle of reins application, it was achieved that this can be done via measuring the pulling the force which is applied by riders. And by having the knowledge that load cells have different kind of application such as compression and tension, another load cell considered for this part of the project. This time a tension load cell which is sensitive against pulling force designed for this application and the schematic view of this design is presented in figure 3.6.

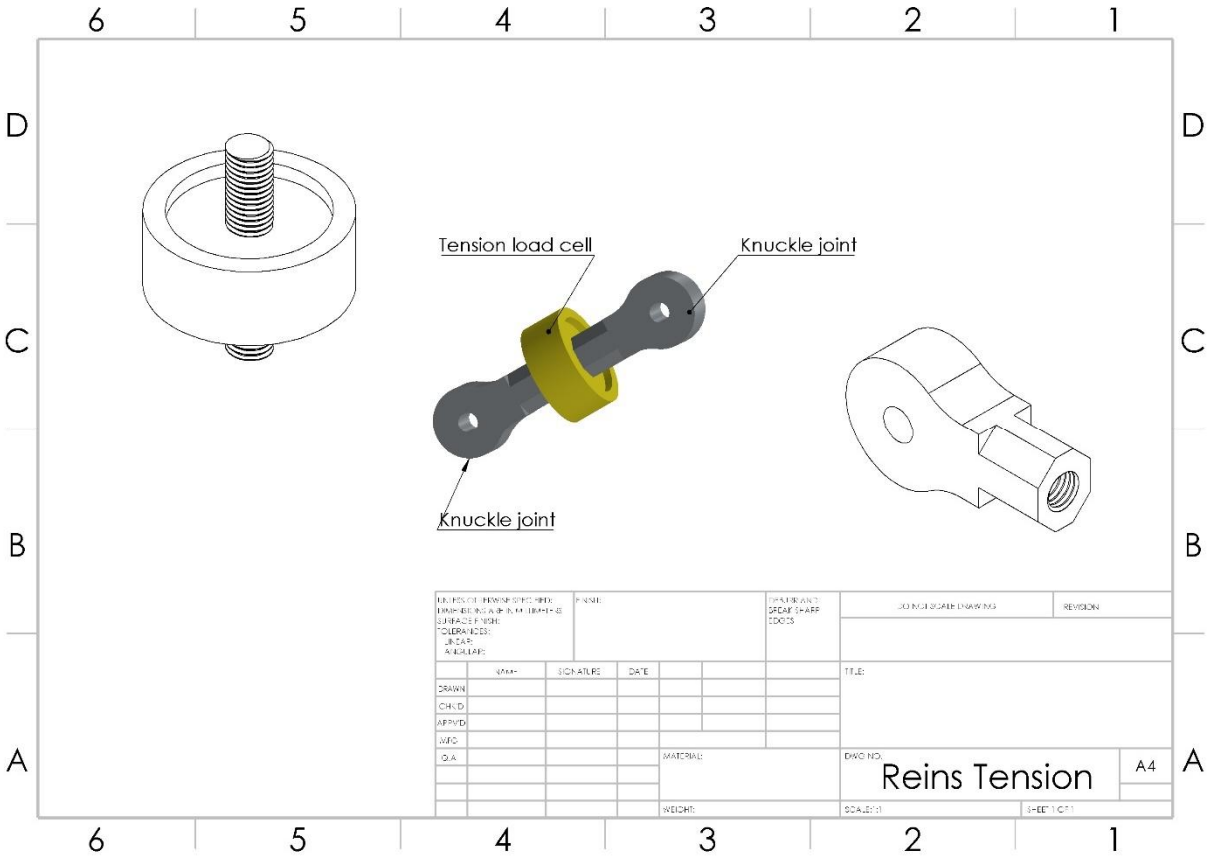

Figure 3.6. Reins tension design.

In this design two knuckle joints are connected to the tension load cell. One of them is fixed to the horse head and reins and another one is connected to the rope which rider will apply the force to it. The operating range of this sensor is 20 Kg. Result of measuring pulling force of human shows that the maximum force that a man can pull is 400N in standing position (Das et al., 2004). By conducting the strength measurement, the result shows that sitting and standing positions have some similarities. But the most important difference of them is in sitting posture (Like what is happening in horse riding) the load that can pull by a person is decreased by 8 to 25%. On the other hand, men pulling strength is decreased 37% in sitting position. Therefore, the suitable range for this tension load cell considered as 20Kg.

3.2 Design of rider simulator interface

3.2.1 Saddle pressure sensor system

The aim of this section of study is introducing a new method of saddle force measurement for using in the horse simulators. One of the important parameters in horse-back riding is the rider sitting pattern in different horse gaits. This pattern recognition becomes more vital when the smooth movement of the simulator needs this measurement as a feedback in the control system. By using this measurement concept, the operation of this simulator can be compared with the real horse. Another parameter that can be study in this invention is the force distribution of riders' body while riding.

In this method riders' weight is considered as a load to the saddle. The saddle will be in the direct contact with the force measuring system that can detect the force distribution. All equipment is installed between two protective, supportive stainless-steel plates. Bottom plate is the surface that can be attached to the main skid of the horse body, while the top plate is the connector of measuring system to the rider's load. In between of these plates, five compression load cells are installed vertically and parallel to the gravity force direction. Due to fast movement of the horse simulator in different patterns of riding, the acceleration is expectable during the riding session. On the other hand, this acceleration causes horizontal shear force on the internal

parts connections, specially load cells. This shear force is an unaligned force which can push one part of load cells in one specific and unknown direction and another part to the opposite direction. Therefore, protection against this phenomenon, is the one of main aims of this invention. In the invention, universal joints are considered for top and bottom part of all load cells. These joints permit more freedoms to the connection parts of the load cells and prevent the load cell from the horizontal shear forces. But considering the measuring direction of the load cells which is vertical and keeping this direction bold beside using the universal joint, leads the invention for using some other parts to guarantee the vertical arrangement of the load cells and whole system. By introducing the linear motion equipment such as flange linear bushing with shaft and holder this idea can be done. These mechanical components are installed in parallel to load cells construction and they guarantee the force applied to the sensors are kept vertically without any rotation or translation.

- *GENERAL OVERVIEW*

In this section, the overview installation of various devices integrated into saddle pressure measurement is presented in Figure 3.7.

The overall dimension of the design is 400x320x157 mm (LxWxH). The design includes 5 loadcells assemblies and 4 linear bearing assemblies. Load cells are supported by means of universal joints. The static tensile or failure load of this joints are 13000 N. The allowable condition variable of the joins is 80000. According to manufacturer's datasheet the calculated condition variable should be less than allowable condition variable in order to have a reliable operation.

Calculated condition variable = Torque (N.m) * Angle(º) * rotational speed(r/min) (Eq. 3.5)

Calculated condition variable = 27.4 * 1 * 2000 = 54800

The calculated condition variable shows that if the maximum allowable torque is applied to the joint which is practically not possible and by having maximum angle of movement which is 1

degree and with maximum rotational speed of 2000 (r/min) which is also not happening in this design, then the value will be much less than allowable condition variable of the joints.

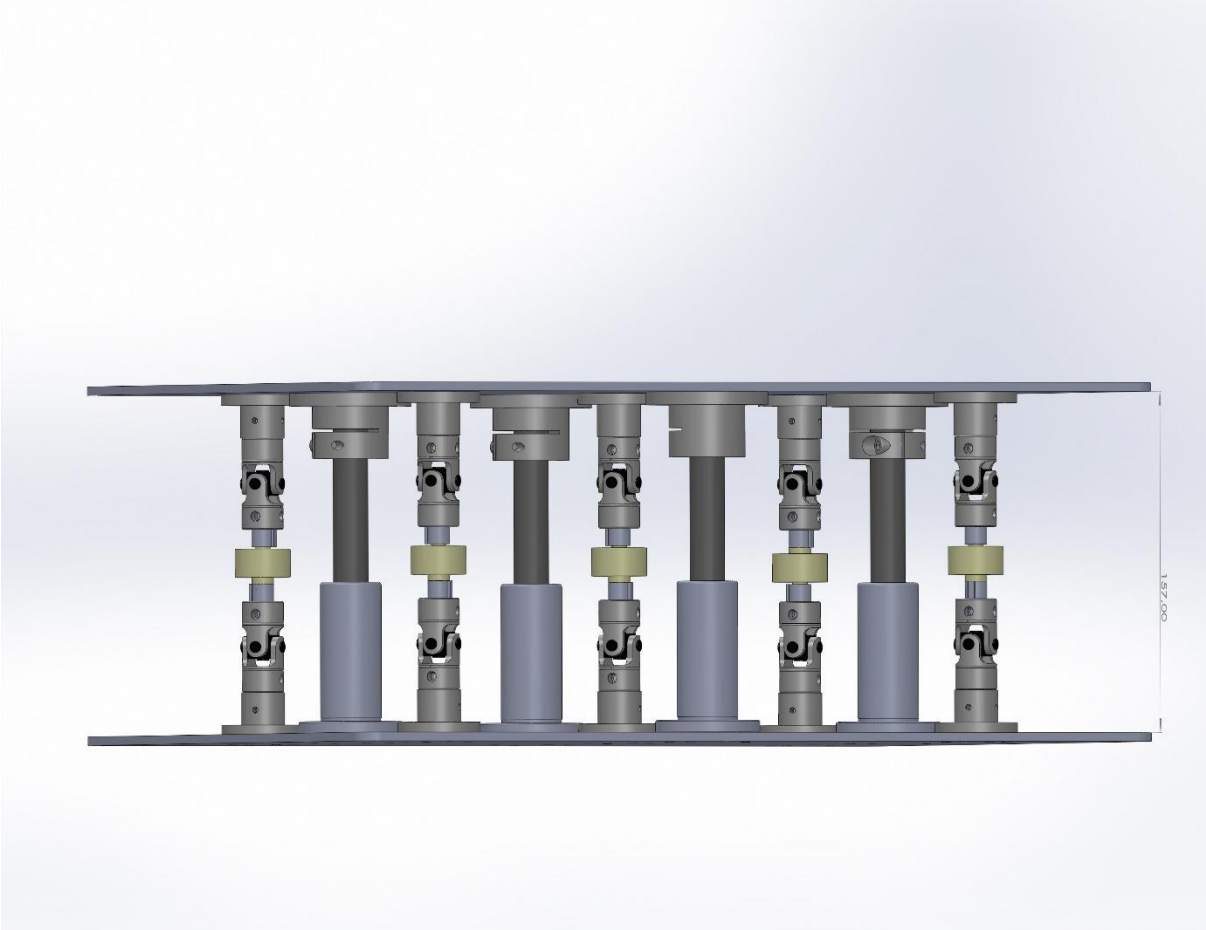

Figure 3.7. General overview (Front view).

In Figure 3.8 the overview shown from top to present the dimension and installation method of two plates in top and bottom of the assemblies.

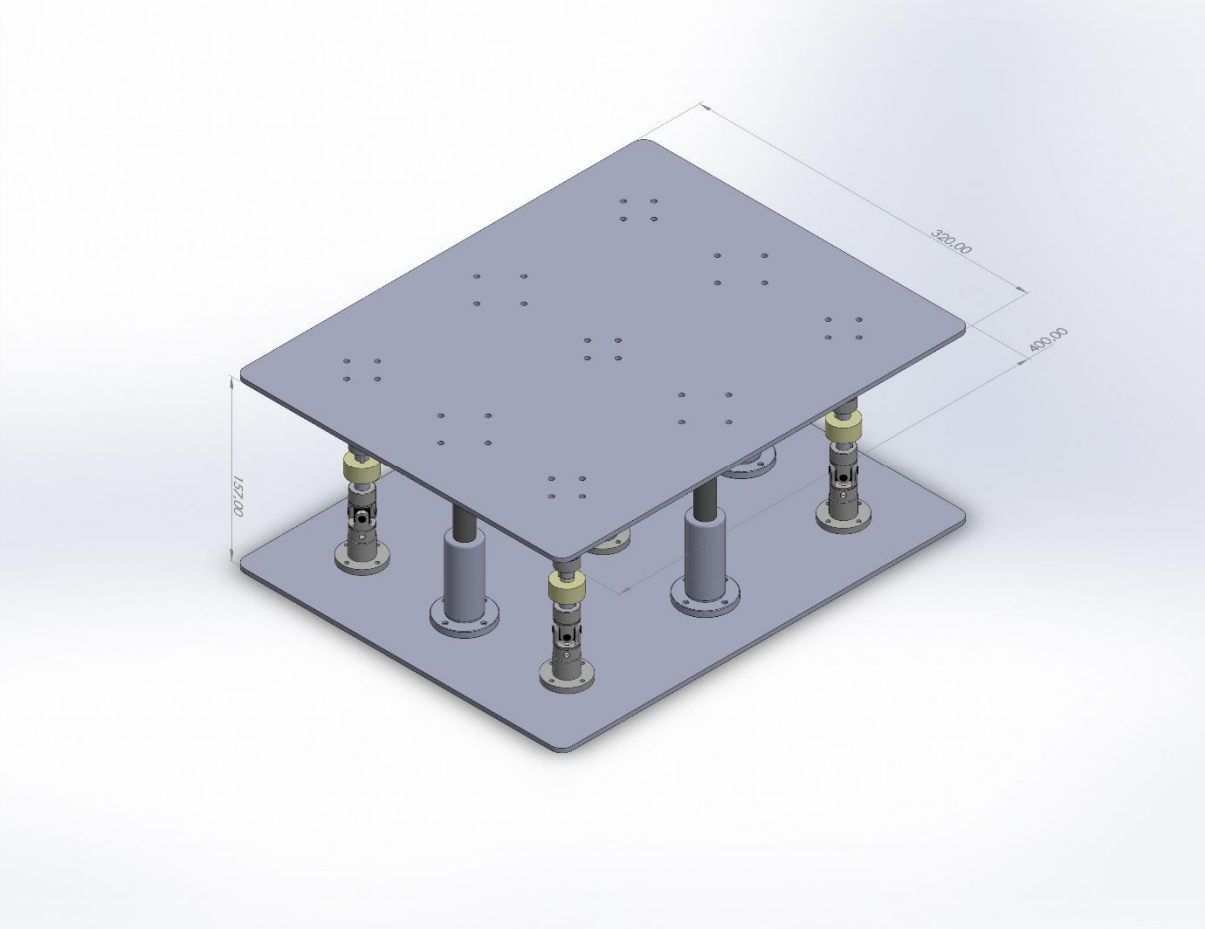

Figure 3.8. General overview (Isometric view).

In Figure 3.9, the installation of all assemblies shown without the top plate. This can help to recognize the space dividing between the load cells and linear bearing constructions. In the design stage, the main aim is to cover all the area of the surface under measurement of the load cells while the linear bearing ensure the linear movement of the structure and also horizontal shear force does not have any effect on the structure.

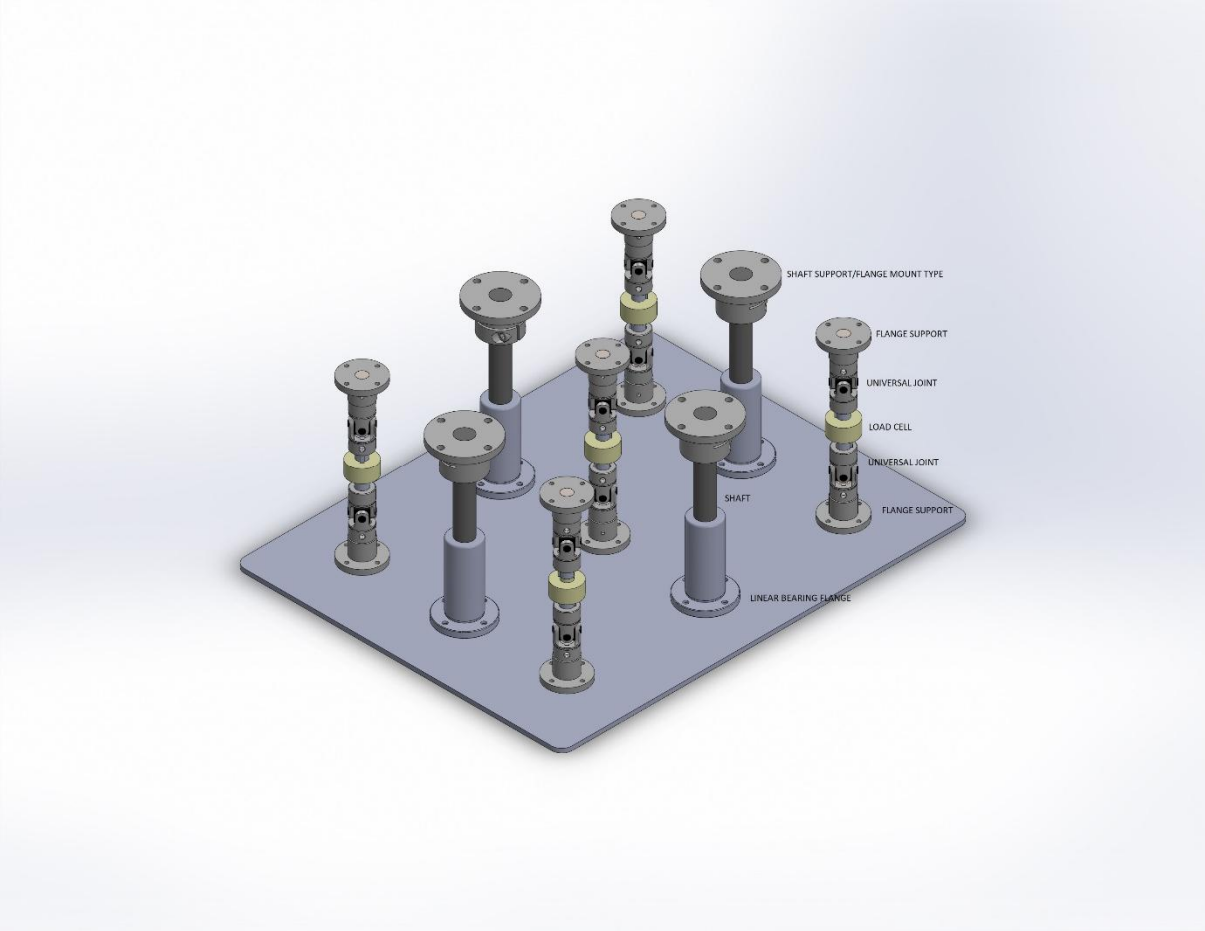

Figure 3.9. General arrangement (Isometric view without top plate).

- *LOAD CELL ARRANGEMENT*

In this construction, there are 5 bars which consist of 5 load cells. These load cells equipped with universal joints on both side of them. Then these components connect to the flange support. All these components material is stainless steel 316. Figure 3.10 shows this assembly. The electrical connection of the load cells should directly connect to the internal electrical junction box. Monitoring and signal presenting should be done inside Beckhoff PLC.

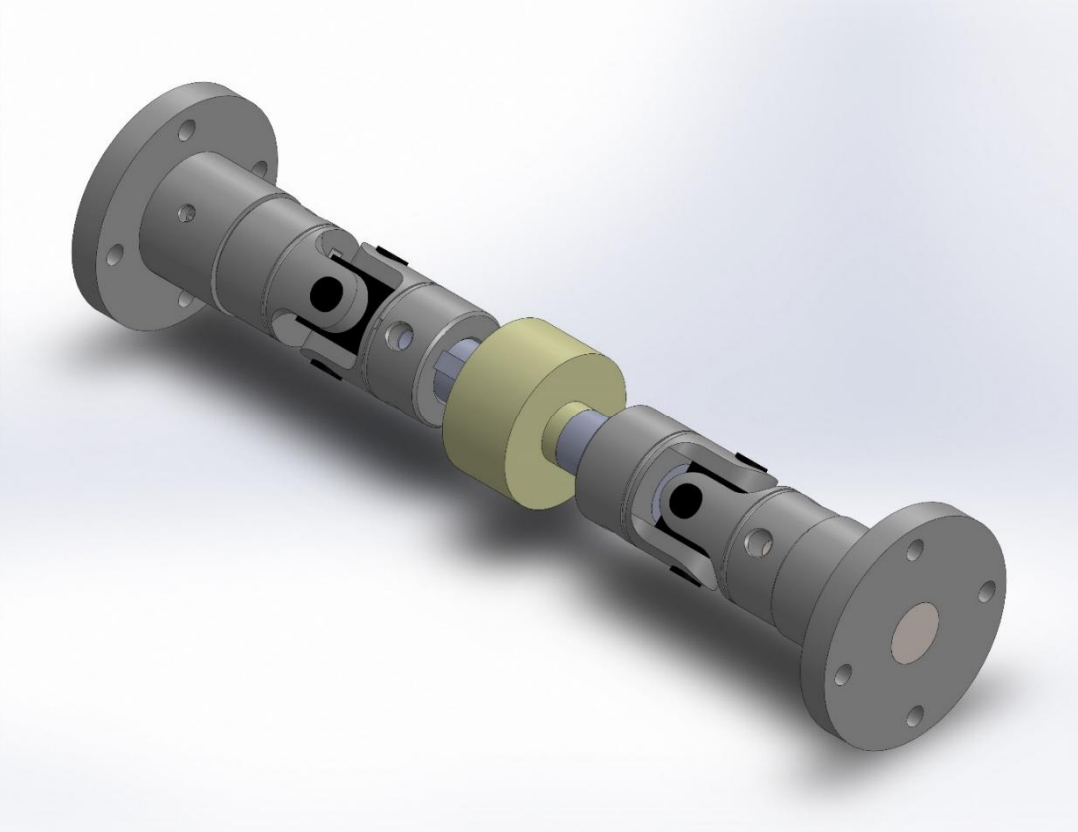

Figure 3.10. General arrangement for load cell assembly (Isometric view).

- LINEAR BEARING ARRANGEMENT

Linear bearing considered as a leg of the structure, while they support the stability of the whole structure. Linear bearing flange consists of many ball bearings which installed inside the flange and will be in a direct contact with shaft. Figure 3.11 shows the internal section view of a linear bearing, which is considered in this design. Figure 3.12 presents the assembly of the linear bearing with the shaft and shaft support flange.

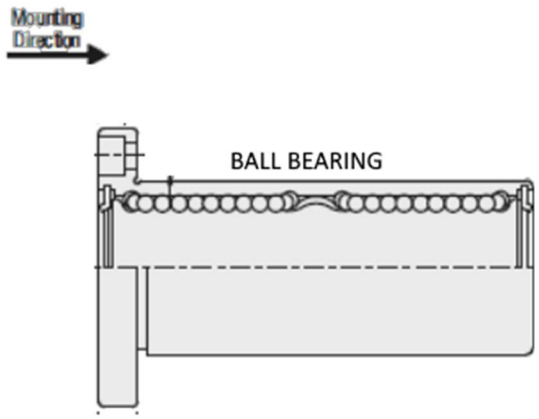

Figure 3.11. Linear bearing with flange connection, cross section view.

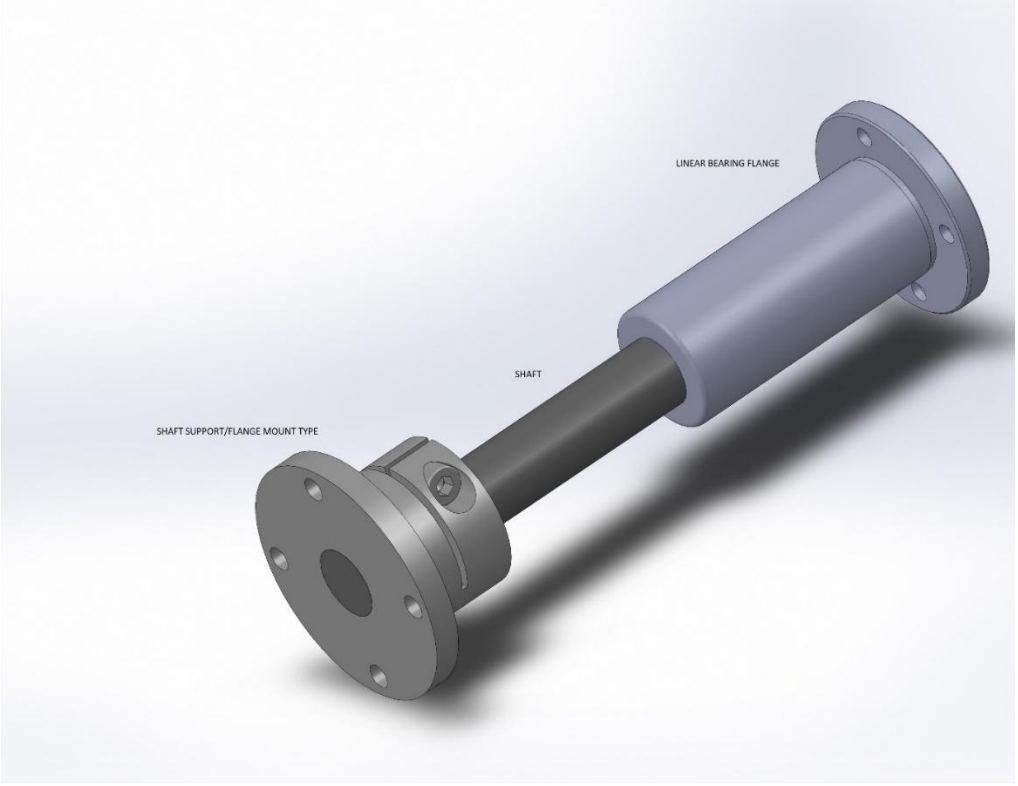

Figure 3.12. General arrangement for linear bearing assembly (Isometric view).

Horse Saddle force measurement system, which is presented in this section, aims to measure, and map the forces that are introduced from the rider's body accelerations during the riding session. Although the size and shape of riders and horses vary in each case, this design brings a general and applicable approach to all the different cases. There are some other types of solution existed and used by many horse simulators manufactures, but this model covered the requirement of the accurate measurement while ensuring the reliable and convenient measuring of the force.

3.2.2 Integration of the calf pressure sensors

The main aim of this section is to describe the design procedure and represent the modeling and experiments which are conducted according to the project scope. Calf pressure design is the first design experiment in this project, this model includes force resistive sensor connected to Arduino for force signal processing. Since this sensor should be integrated in chaps and by having the flexible force resistive sensor, installing of the sensor is possible on the surface of chaps. Although that the sensor manufacturer recommended to avoid rolling and use the sensor with flat installation, but the experiments show that the deformation of the sensor by means of installing on chaps' surface is not make any error for its force detection. Therefore, the internal part of the chaps considered for installing the force sensor. In addition, the closest place for installing the electrical box is on another side of the chaps. This electrical box basically consists of Arduino and battery for supplying the required voltage to it. A protective elastic cover is recommended for holding all the equipment and provide a better overview to the chaps. The wiring between electrical box and sensors can be done in the bottom of this protective cover. Since the force resistive sensor has two connection ports, it needs two wire going out from electrical box and connect to it. So, in this case the low amount of wires is an advantage in case of assembling and future diagnostic. A pull-down resistor which is required in wiring of this sensor can be installed inside the electrical box.

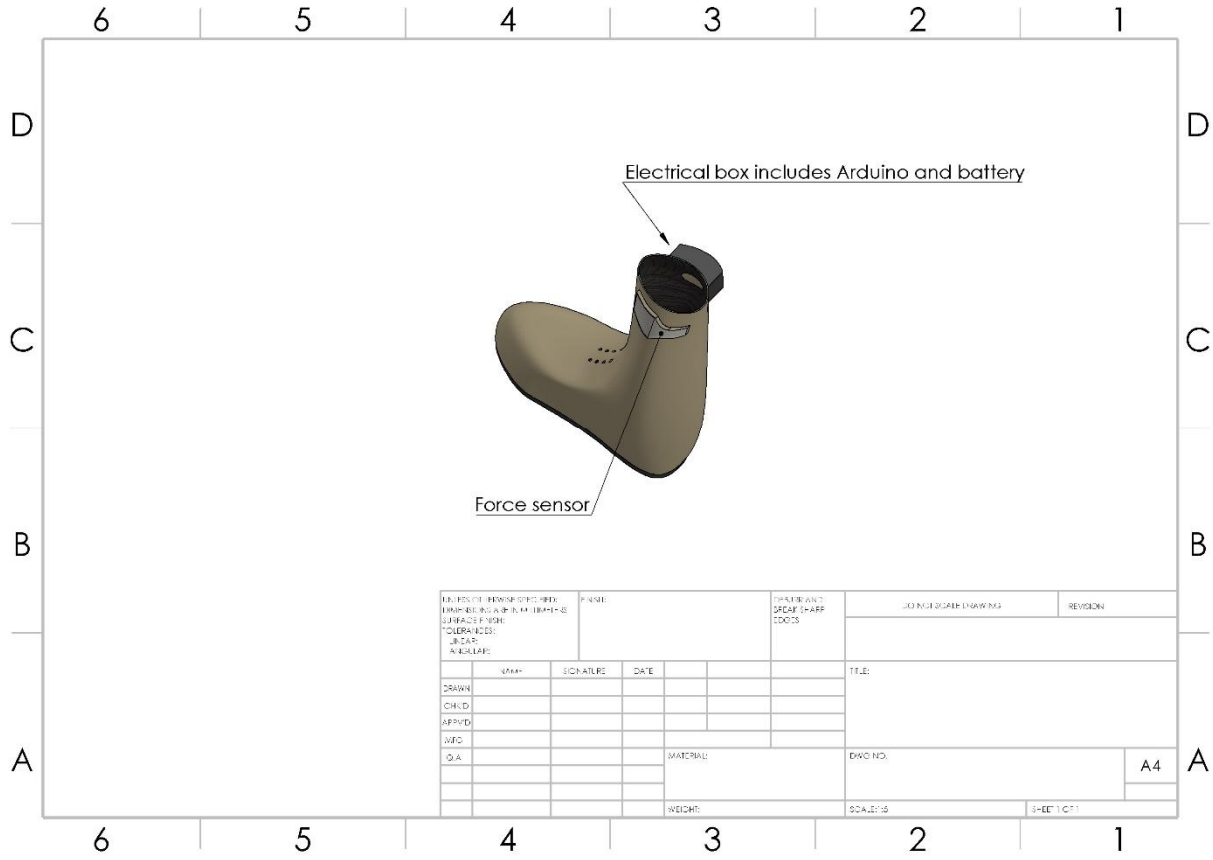

Figure 3.13. Force sensor and electrical box integrated into chaps.

By having this prospective, the integration between these two main components was started. The first study was the interconnection and wiring diagram of them. Reading the value of the force is another area of concern in that experiment. Carrying out researches shows that resistance of the sensor versus force applied to it, are in opposite. It means that by increasing the force, resistance will decrease. In other words, resistance is inversely proportional to force. Figure 3.14 shows the behavior of the sensor by applying force to it.

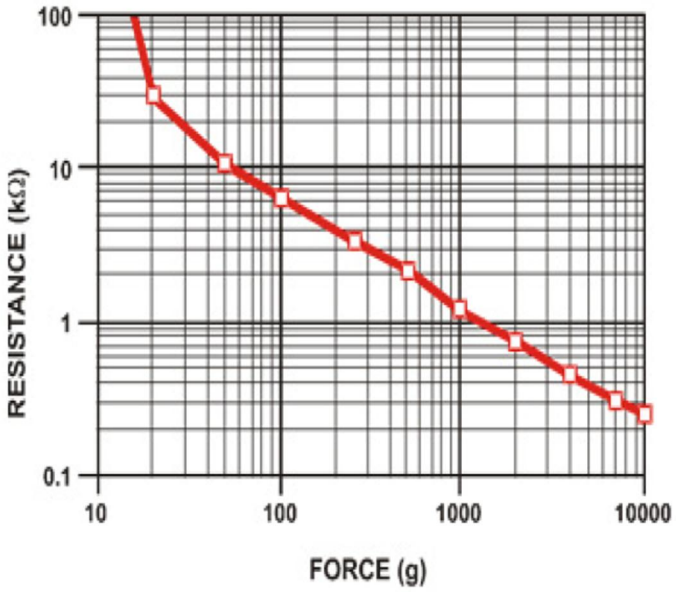

Figure 3.14. Force vs resistance chart. (acc. to Manufacturer's datasheet)

Furthermore, a voltage divider circuit should consider for applied force measurement. This circuit can be provided by another resistance which installed in series with the force sensor. The aim of using this circuit is t provide variable voltage for the Arduino analog input pin. This is the measurement method of force. In Arduino by using the analog to digital internal block, it can read the value of the force. Below equation shows how to calculate the force by changing in the voltage.

$$V_{out} = \frac{R_M * V+}{R_M + R_{FSR}} \qquad \text{(Eq. 3.5)}$$

R_M : pull down resistor, series to the FSR

In this equation the output voltage which is measured is the drop voltage around R_M . By using a 10 KΩ as a pull sown resistor and 5V, the result is as bellow:

$$V_{out} = \frac{10\ K\Omega * 5V}{10\ \text{K}\Omega + 100\Omega} = 4{,}9\ \text{V}$$

After the fundamental studies about the sensor, electrical wiring diagram is carried out.

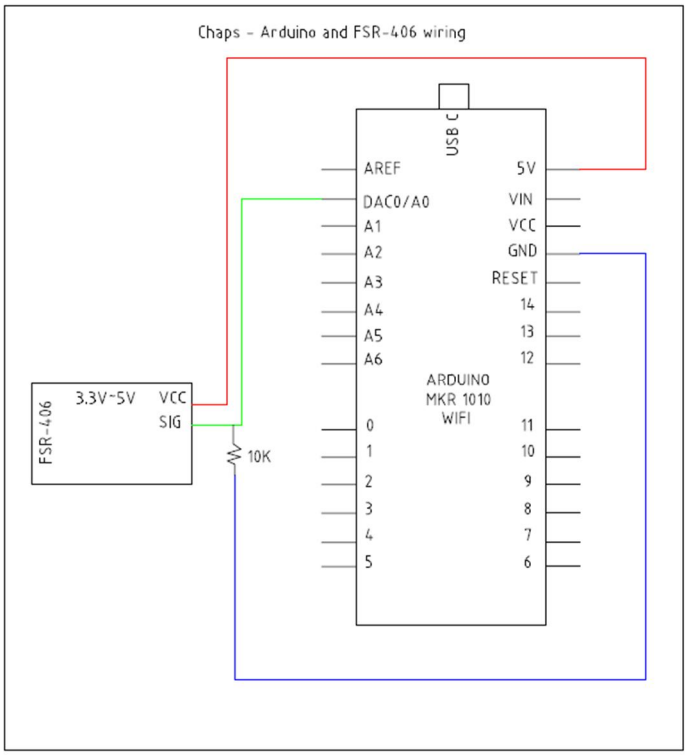

Figure 3.15. Electrical diagram; FSR integrated to Arduino.

In this experiment, Force sensor is connected to the pin number 0 which is the first analog input pins of the Arduino. It is also equipped with the ADC (Analog to digital converter) capability. This can help to convert variable voltage from FSR convert to digital signal and present in the serial monitor section of Arduino IDE. The variable voltage generated from the series connection between FSR and 10K resistance. The program is written for the monitoring the value of the force is presented in figure 3.16. for making the output more visible, LED connected to the pin 11 of the Arduino. This LED turns on and off in case of sensing force by the FSR. The baud rate which is the serial data transmission between Arduino and computer is set to 9600. The operating range of force detection divided into 5 sections. When there is not any force sense by sensor it shows “No pressure” If the pressure is less than 10, 50 and 150, It generates “Light touch”,” Light squeeze” and “Medium squeeze” respectively. Finally, for the forces more than 150 the message of “Big squeeze” represent in the serial monitor output of the Arduino IDE. And the mapping of the input signal scaled between 0 to 1023 for changing the brightness of the LED. By this experiment force signal transmitted and measured.

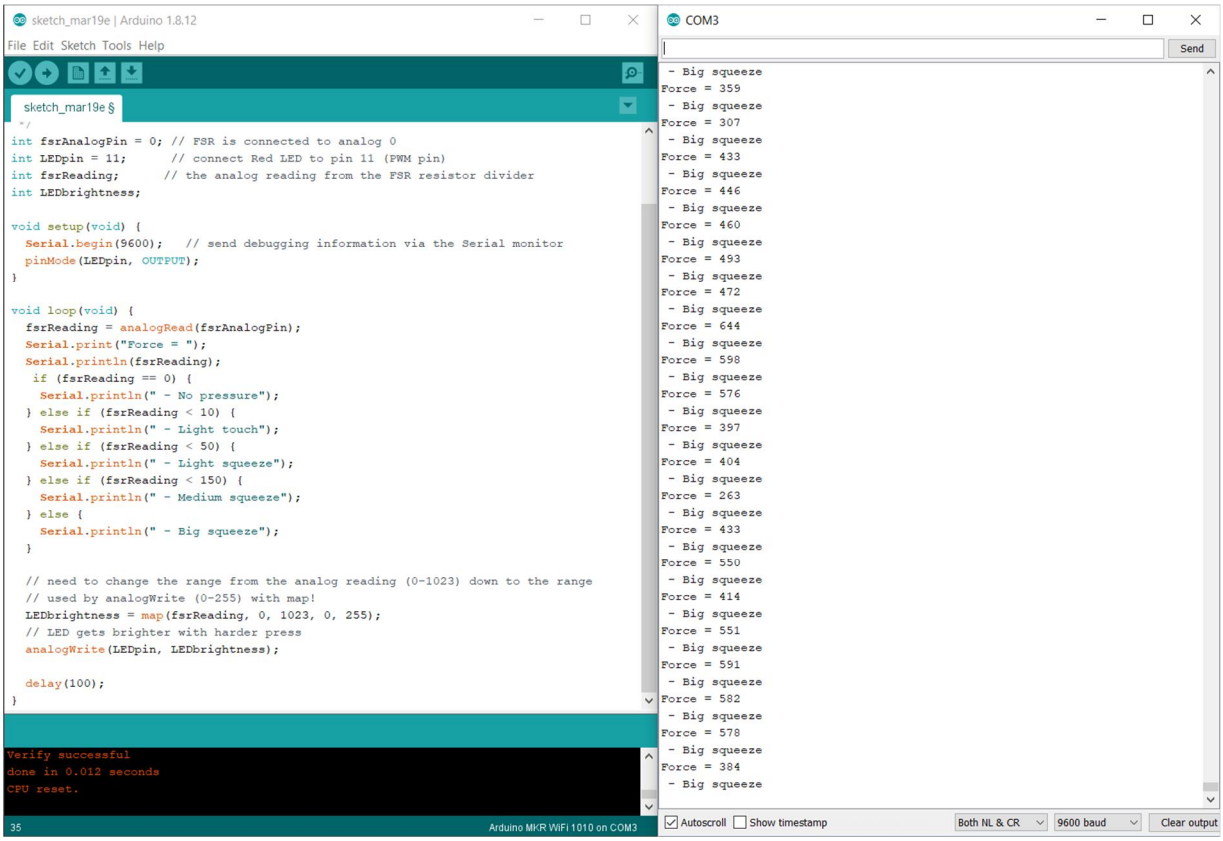

Figure 3.16. FSR code executed in Arduino IDE.

The experiment with FSR shows that, this sensor can work appropriately in long time use without decreasing in accuracy and response time. For future works these messages which are generated from force sensor can be scaled in Beckhoff PLC. Moreover, by writing the logic code for this signal, simulator can be able to recognize the aim of the rider, whether to aim for a light start or changing to another pattern. It is also possible to add another force sensor to this circuit. If the rider feels that it is not enough for sensing the force and more area of sensing is needed, another force sensor can be installed in series with the existed one and increase the force sensing area. After doing this experiment, components are mounted and integrated to the chaps. Figure 3.17 present the integration of the FSR and Arduino to the chaps.

Figure 3.17. FSR and Arduino integrated to the shoes.

3.2.2 Inclinometer integration

In this section the inclination of the rider's calf is measured. As mentioned in section 3.1.3, angle sensor which used is Murata SCL-3300-D01-PCB. This sensor has capability to measure the inclination of three axis (X, Y, Z) and by using the angle calculation equation, it can present the angle via digital SPI interface. This inclinometer also equipped with other sensors and it can show the temperature and acceleration separately. SPI interface consists of four main signals such as:

- CSB (Chip Select): This signal allows the communication between master and slave. In the beginning of program this pin should be trigger to high and then it should set to low. All the data acquisition between master and slave is carry out when this signal is set to low.
- SCK (Serial Clock): It is a serial clock pulses which is generated by the master device and it is used to specify how much the master's clock should be divided in order to the sampling signal can be recognized.
- MOSI (Master Out Slave In): data which is the output from master is transmit to slave by this pin. Slave read this data from master.

- MISO (Master In Slave Out): slave data is send to master by this pin and master read the data from slave.

This sensor should install in the top part of the chaps and connects to the Arduino via six wires. The SAMD21 processor of the Arduino allows one SPI communication. In the Arduino there is not any pin defined for CSB connection by default, so it needs to define this pin in the code. The connection diagram of both sensors is presented in figure 3.18.

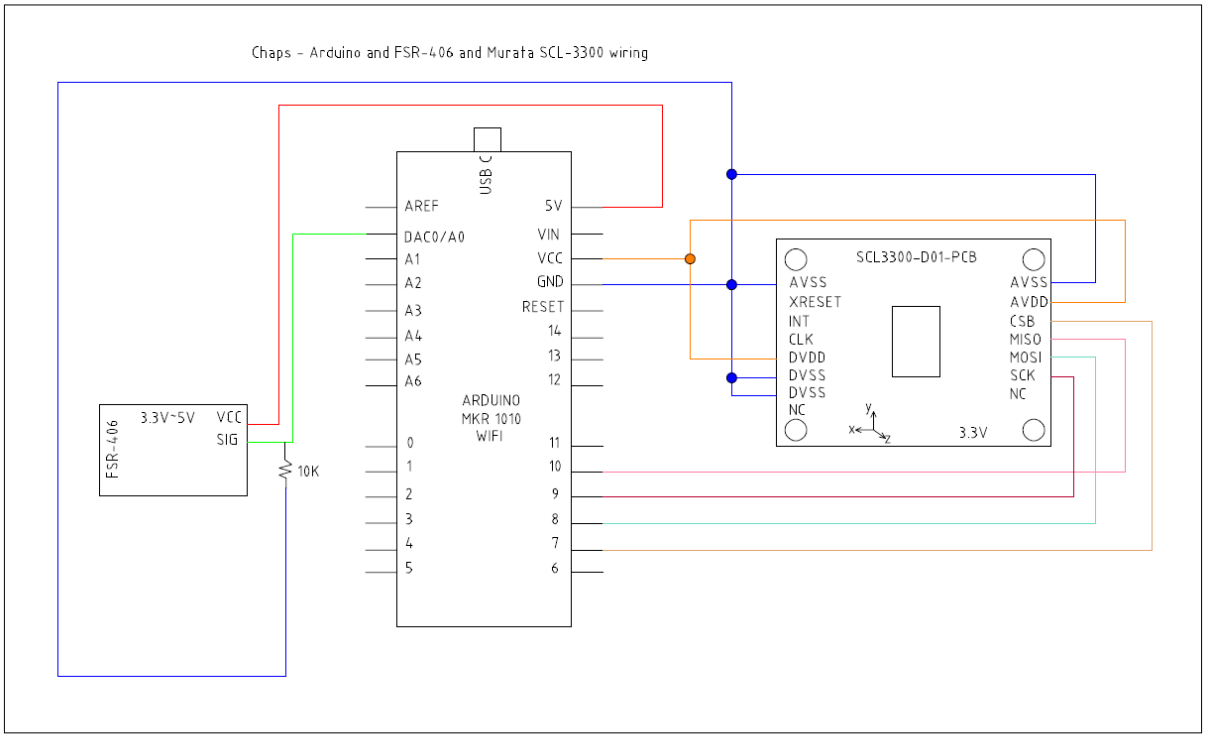

Figure 3.18. Force sensor and inclinometer integration into Arduino.

After wiring the sensor to Arduino, the start up procedure of the sensor should be considered. By applying 3.3 V to the “VDD” and “DVIO” pins of the sensor, it will start reading the memory and setting the signal path. It takes around 10 ms for doing this step. Next step is to define the measurement mode of the sensor. It has four measurement modes. The first two modes are used for acceleration and last two modes are used for inclination measurement. But mode 4 is the

selected mode for this measurement since it is based on low pass filter and includes low noise capability. After that sensor will clear its status summary and read new status summary and SPI response is provided in this step. Then the it provides RS (Return Status) signal for indicating the successful startup of the sensor. In the final step “ANG CTRL” signal will write in SPI and specifically in MISO pin of the sensor. In mode4, the sensitivity of sensor is 182 °/g, The method of calculating the angle is to 1g * sin(Θ). In this equation, Θ is the inclination angle with respect to 0 g position. By considering this method and also the specification of sinus function sensitivity which is reversely proportional to tilt angle. The SPI data transmission is started when falling edge detected by CSB. Sensor’s output is generated by detecting the rising edge from SCK pin. Finally, this output is transmitted to the Arduino and it is presented in figure 3.19.

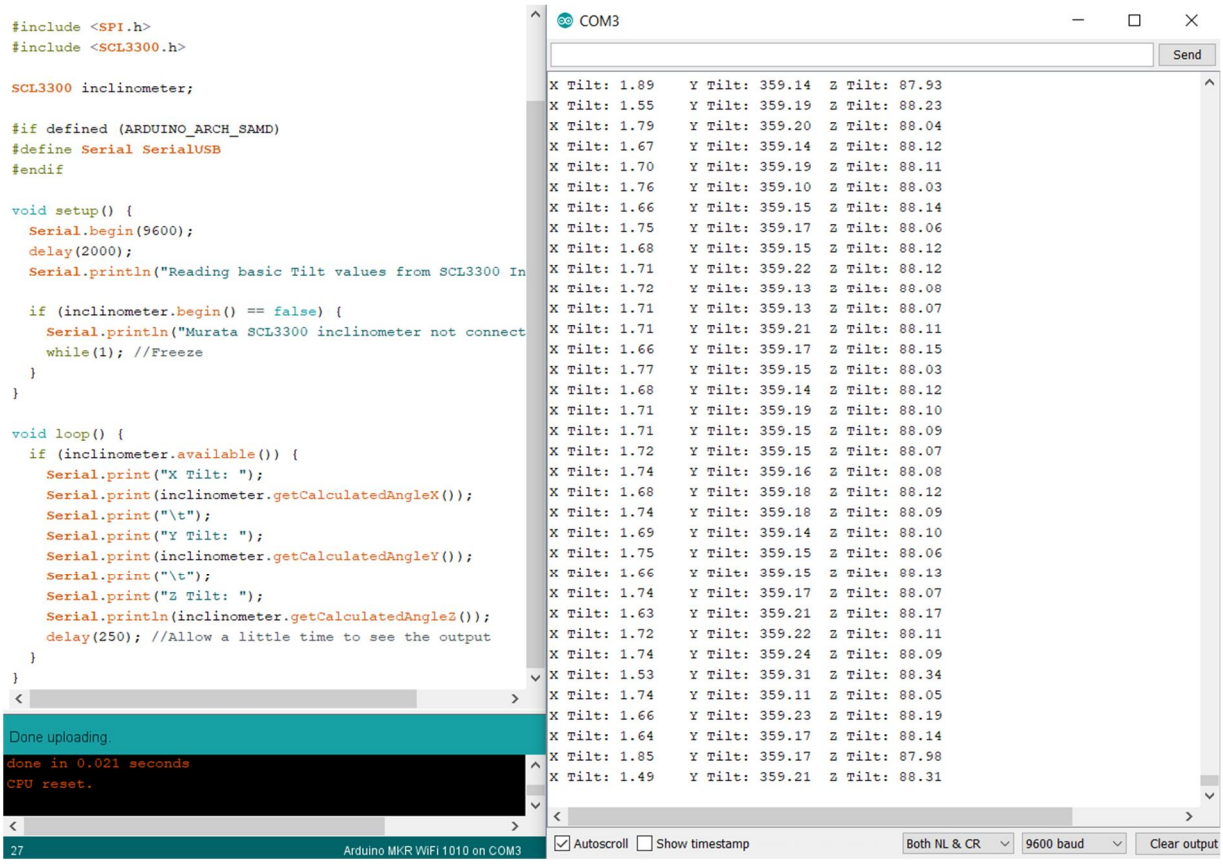

Figure 3.19. Inclination measurement’s result in Arduino’s serial monitor.

3.2.3 Wireless solution design and integration

- *INTRODUCTION*

Wireless solution is the most convenient approach for integrating components to each other. This method can bring reliability, freedom and reasonable cost saving to each project. Wireless capability can improve the data communication speed and transfer them quicker from sender to receiver. Availability of this communication is another reason for using them. Since wireless technology lets users to stablish the connection even when they are walking or doing any other activity. In this study, using wireless technology is a most. Because the rider should have enough freedom to act on a simulator. Rider's should not worry about the wires and cables which may connect to components. The simplicity in the configuration of the wireless networks is another reason for migrating to this solution, reducing cable and easy installation can decrease the time of prototyping and fabrication. By having the possibility to integrate wireless network with internet, the number of digital workplaces and projects increased significantly. After investigation about possible options for wireless network communications, MQTT protocol selected for implementing the data transferring base on it. The connection aim is to transmit force sensor data from Arduino to the Beckhoff PLC. And by having this knowledge that both components are supports MQTT protocol, the integration experiment started.

- *MQTT presentation*

MQTT is communication protocol based on publishing and subscribing. It can prepare safe, bi-directional communication between internet-connected devices such as sensors, controllers, actuators. this protocol enables us to collect telemetry data from multiple devices, also store or analyze the data. controlling the process from phones and tablets are also possible in this method. Messages transfer between applications. Individual applicants can subscribe or publish to a specific topic according to their access to that topic. Message broker plays an important role in this protocol. It decouples receiver and sender. The basic requirement for this communication is that receiver and sender know their destination and respective address. generally, message broker handles the distribution of messages. Figure 3.20 shows the basic principle of the MQTT structure.

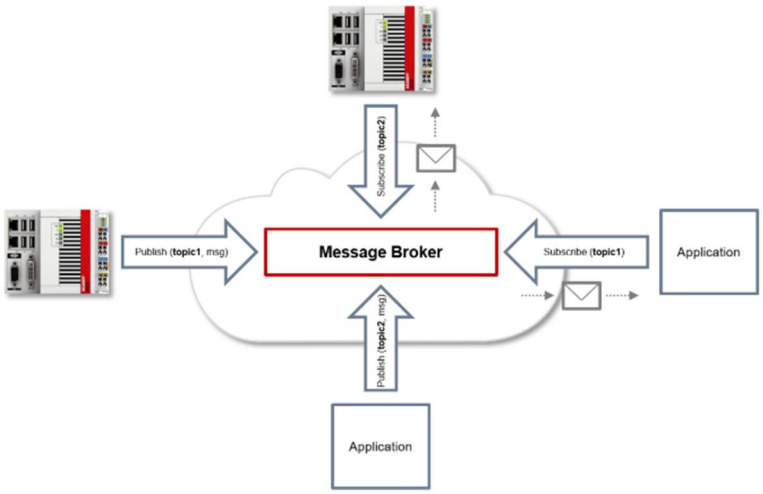

Figure 3.20. MQTT structure (Beckhoff Technical Data).

by using the message broker based on MQTT protocol, publishing and subscribing tasks can be done.

QoS (Quality of service) is another field of this network protocol that should be considered. This concept mainly focused on guaranteeing of the message transferring between publisher and subscriber. In this case, MQTT features three different levels of guarantee:

- 0 – Not more than once
- 1 – at least once
- 2 – exactly once

In both publishing/subscribing communication QoS must be considered by message broker separately. the publisher sends the message with respective QoS to the message broker and subscriber uses that QoS level that was specified when the subscription was established.

In level 0, the receiver does not acknowledge the receipt, and just one message sends. In level 1, the system will guarantee that the message arrives to the receiver at least one time. But sometimes it is possible that the message arrived at the receiver more than once. In this form a PUBACK message will send from receiver, which means that it receives the signal. if the PUBACK message become fail to arrive in a certain time this will lead to resending the message to the receiver. In level 2 the system will guarantee that the message arrived at the receiver exactly one time. It will happen via handshake mechanism in MQTT. this procedure makes this

level the safest one but also the slowest as well. In this level receiver acknowledge the receives of signal through PUBREC signal. this will keep in sender of the signal until it receives the PUBCOM signal. This method can prevent against duplicate send of message. Whenever one of the messages become lost in a certain time the sender will publish the last message in the memory.

- *ARCHITECTURE*

Control system architecture defines the whole integration and communication between controllers, sensors and their connection methods. In most applications, requirements of the control system specified with level of the complexity, functionality, and safety of the system. The main controller in this study is Beckhoff CX-2030 which is equipped with LAN connection. By connecting the CPU to the router, then it will be able to connect to the wireless network and message broker. Therefore, it will have access for subscribing to the MQTT topics. In Arduino side, force sensor is connected to Arduino via analog input pin of the Arduino and a 3.7 V. 3000 mAh battery supplied it. The FSR sensor needs a pull-down resistor that can be installed in Arduino electrical box. Angle sensor is connected to Arduino via SPI communication protocol. It includes four signal wires such as CSB, SCK, MOSI and MISO. Also, two separate wire for means of power supply should be considered for this sensor. Angle sensor should install in a separate electrical box and the installation method should be in parallel with the ground surface and perpendicular to gravity vector. One of the differences between these two sensors are the level of their required power supply. While FSR sensor is connected to the 5V supply and the signal is processed with this voltage, the angle sensor is supplied by 3.3 V which is accessible via Arduino. Arduino automatically derive this voltage and the angle sensor should be connected to the “VCC” pin of the Arduino. In this case the both “GND” connection are the same for both sensors. Figure 3.21 presents the architecture of the integrated system.

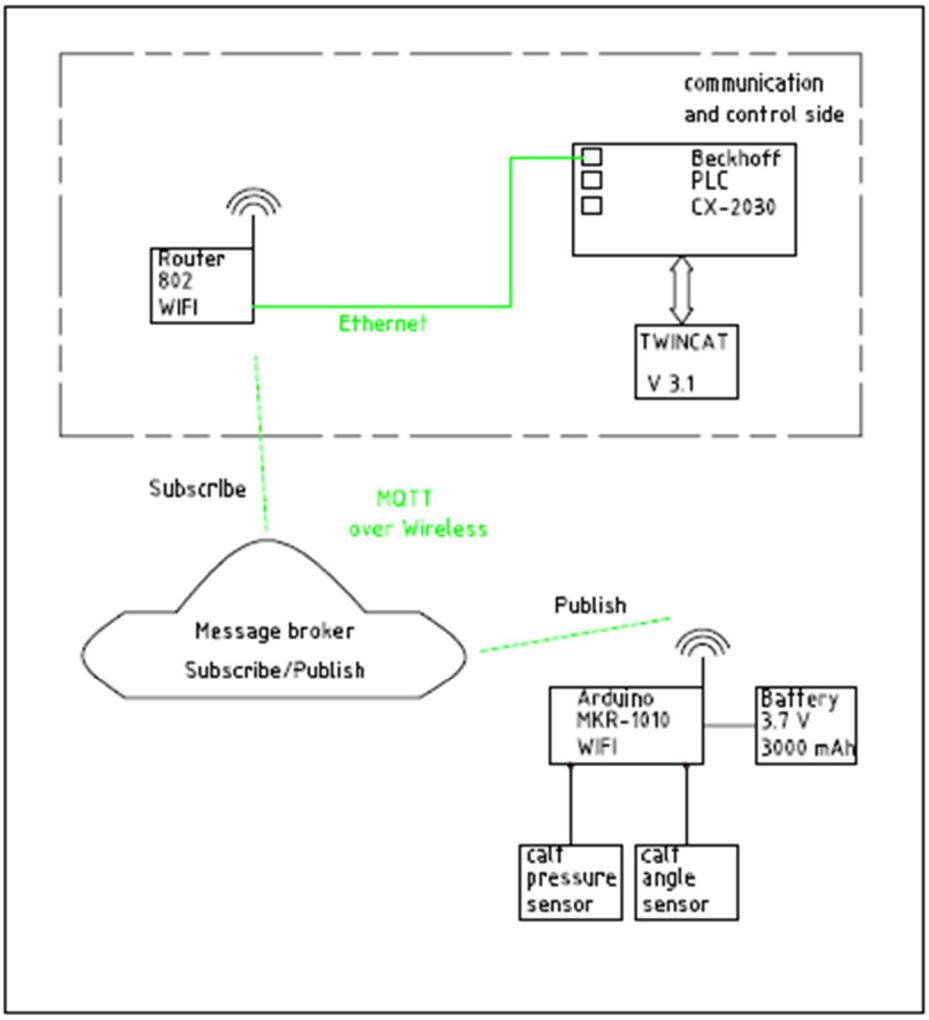

Figure 3.21. Control system architecture.

- *MQTT experiment over internet*

The first experiment of transmitting data to the cloud conducted with www.flespi.io website. This website operates as a message broker and messages subscribe or publish to/from devices by entering the website name and token address as well as topic name. Most of MQTT brokers website are not free and they have a monthly subscribe charge, but this website is completely free, and it provides a reasonable performance comparing with other providers. This broker could connect up to 200 devices in MQTT network. Moreover, there is not any functionality limit in all their services. The first step of working with this broker is to create a free account

and then create a test channel. This test channel provides a token which is the key for connecting the devices to this broker. There is another important parameter in this connection that should be considered such as host name, in this case “mqtt.flespi.io”.

Port type basically divided into two categories:

- MQTT over TCP: 1883 (TLS) or 8883 (SSL)
- MQTT over WebSockets: 80 (TLS) or 443 (SSL)

TLS and SSL are two transport encryptions which use a handshake method to transmit different parameters to generate a safe connection between server and client. When the handshake is accomplished, the encrypted communication of server and client is fulfilled, and this solution can guarantee that no attacker cannot spy on the data of the communication. Since all computer networks needs to have a security layer for their application such as instance messaging, web browsing and VoIP, Using of TLS become more vital for this specific application. This protocol is mostly used in client-server applications. In this experiment, TLS method used for having this level security in communication. TCP/IP is the protocol that the data should transmit on it. In this process there are many devices that data packets should pass through them such as routers, exchange points and firewalls before reaching to the destination of the target. All these devices and points can have access to the data and change or modify them if the suitable protection layer does not consider. The security aim is to prevent reading the data by mentioned points and devices. TLS method can ensure the data content is not accessible by anyone else except end target. MQTT communication executed over TCP protocol. And encrypted communication is not used by TCP by default. On the other hand, providing data integrity and privacy is another preliminary goal of the TLS method. Internal layers of TLS consist of TLS handshake and TLS record protocols. The encryption and decryption between client and server are conducted by means of a session key and it will continue up to the communication is closed by one of them. Therefore, a TLS security layer is required for the communication based on TCP. Figure 3.22 shows the established connection and the result of the connection of Arduino with the flespi.io MQTT broker.

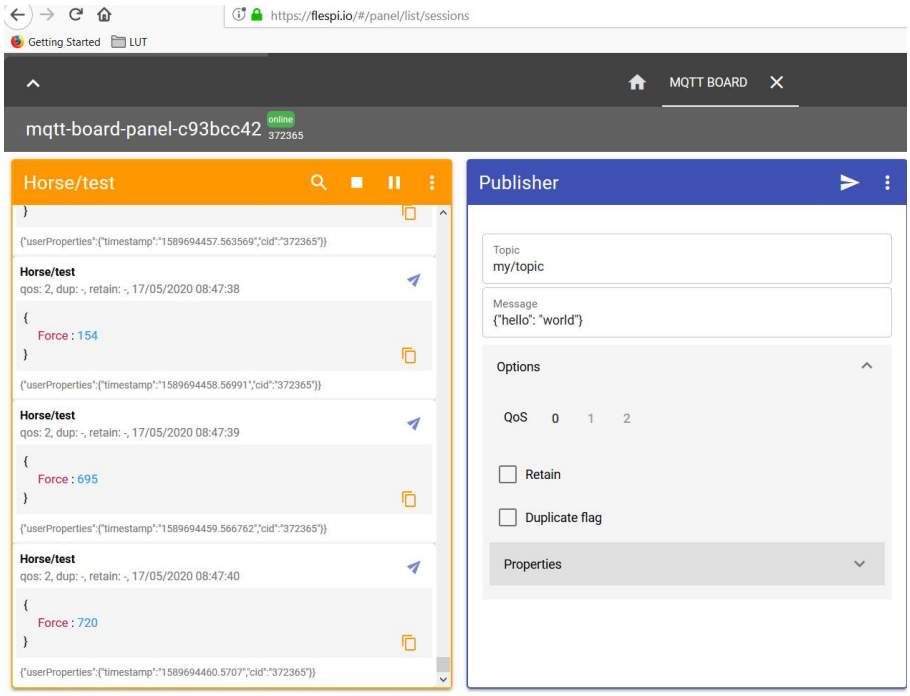

Figure 3.22. MQTT connection over internet by using flespi.io.

In this experiment force value, which is measured by force sensor is transmitted via wireless connection of Arduino to the internet and the message broker website. The data shown in the subscribe panel and accessible just by the end user of the communication which is defined in the program.

- *MQTT experiment without internet*

Although that a powerful security layer applied to the communication packet in the previous experiment by internet, but it is always a risky situation for making the decision to use the internet or not. There is a safety consideration about disconnection of the internet by an accidental reason which is predictable in all places. Sometimes lack of the updates and new methods in hacking and attacking is dangerous too. The second experiment of MQTT connection by using a third-party software “MQTT.Fx” which is also free and simple for using. This software is work like a MQTT client. It can subscribe or publish to a special topic. In most of laboratories use this software as a MQTT simulator and test their connection first with this software. One of the requirements of this software is to install “Mosquitto broker” before it.

Mosquitto broker can be found in windows services. It is important that the startup method of it set on automatically after startup, otherwise the connection will not establish. But back to MQTT.Fx, it is a java-based FX client of MQTT. After installation and by selecting the local mosquitto, it will automatically connect to the broker which installed previously. By selecting setting, the token can be generated by the software and it should be exactly used in the Arduino code. Also, the broker address should be set to 127.0.0.1 for connection as a local host. The final step is entering the name of topic which the software should publish/subscribe to it. There are some other tabs such as “User credential” which is considered for entering the username and password of the domain. The username section should be filled with the device name followed by the “&” character with product key.

Structure: ${DeviceName}&${PrductKey} Example: FSR406&fghj

In Arduino side also some changes in the code is required. Token in both sides should be the same and MQTT server address should be changed to the IP address of the local host. After applying these modifications, in subscribe tab of the software it is necessary to type the topic name which the software wants to subscribe to it. Then by pressing the connect button, MQTT.Fx will become a subscriber to that topic. The same thing should be happened if the publisher option is required. After subscribing the messages from Arduino shown in the bottom of the software. There is a “Log” tab in the MQTT.Fx which enable users to view the operation logs and review the errors that may happened during the connection. Figure 3.23 shows the result of the experiment after establishing the connection between Arduino and MQTT.Fx without internet connection, by using a wireless router.

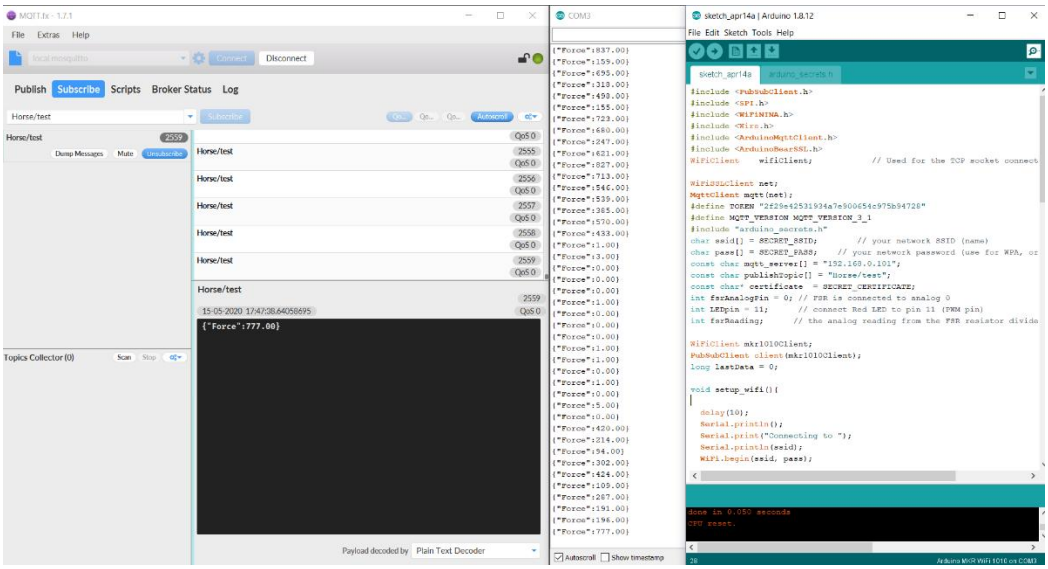

Figure 3.23. MQTT connection without internet by using MQTT.Fx.

- *MQTT experiment with Beckhoff TwinCAT*

By gathering the knowledge about this protocol and having two experiments with other platforms, the main experiment for connecting the Arduino to the Beckhoff PLC software is conducted. TwinCAT software is an open-source software which brings real-time controller experience to each PC. This software is based on “IEC61131-3” standard. It means that all programming languages such as (IL, SFC, FBD, ST, LD) is executable on this software. It is designed for programming and systems run-time monitoring on PC or either on PLC. The engineering version of TwinCAT has the feature to integrate in “Visual Studio” and it supports all visual studio’s interfaces. There are also some integrated features in TwinCAT that makes it more valuable in the industry. Integrating to Matlab/Simulink, C++ debugger, configuration of the destination system, scope view and bode plot integration are other extensions of this software. The installation process of TwinCAT which is going to be integrated into Visual studio is an important procedure. All the steps should be executed very carefully, otherwise it leads to serious errors that the solutions of them are rare and can not be find in resources. First the Visual studio should be installed. In the installation wizard some extension such as “.NET desktop

development", "Desktop development with C++" and "Universal windows platform development" need to be checked because the execution of TwinCAT is depend on installing them. After finishing the visual studio installation and by running the TwinCAT XAE which is the engineering version of Beckhoff, it is automatically recognize the visual studio shell and install the TwinCAT software integrated to Visual Studio. After finishing the installation, three specific menus will appear in the Visual Studio, PLC, TwinCAT and TwinSAFE. These menus and the TwinCAT icon in the taskbar of the windows show that the both software are installed correctly.

Table 3-3 *MQTT variables defined in the code.*

Name	**Data type**	**Initialization**	**Description**
fbMqttClient	FB_IotMqttClient		Function block include in the Tc3_IotBase library
bSetParameter	BOOL	TRUE	Setting the parameter once after execution
bConnect	BOOL	TRUE	It is true if the connection exists between broker and client
sTopicPub	STRING(255)	"MyTopic"	Name of the publish topic
sPayloadPub	STRING(255)		Declare the publish data
i	UDINT		Execution sequence
fbTimer	TON	(PT:=T#1S)	On delay timer
bSubscribed	BOOL		If true it will receive the messages
sTopicSub	STRING(255)	"Horse/test"	Name of the subscribed topic
sTopicRcv	STRING(255)		Name of the subscribed topic
SpayloadRcv	STRING(255)		Declare the subscribed data
fbMessageQueue	FB_IotMqttMessageQueue		Function block include in the Tc3_IotBase library
fbMessage	FB_IotMqttMessage		Function block include in the Tc3_IotBase library
hrErrorOccurred	HRESULT		Shows the error if occurred

The MQTT code for establishing a connection consists of two POU. First POU called “MAIN” which is the main part of the program that all other function blocks should execute inside it. Another POU “PrgMqttCom” is integrated in MAIN and executed continuously. Table 3.3 presents the variable and their descriptions which are defined for the MQTT communication. In this experiment after creating a new project in TwinCAT, in the library section of the code it is necessary to select and insert “Tc3_IotBase” as a reference library, otherwise the program cannot execute properly. In this library there are some prepared function blocks which assist the main program for establishing the MQTT communication. In conditional section of this function block, the MQTT client from TwinCAT is executed and by do not having any error the procedure moves to the next step. In the next step program is checking if the message is existed according to the topic name and there are not any other messages wait in the queue from past, then it will start to show the messages. The messages in this code are shown in the payload variable. Figure 3.24 shows the code for execution of the communication.

```
fbMqttClient.Execute(bConnect);
IF fbMqttClient.bError THEN
    // add your error logging here
    hrErrorOccurred := fbMqttClient.hrErrorCode;
END_IF

IF fbMessageQueue.nQueuedMessages > 0 THEN
    IF fbMessageQueue.Dequeue(fbMessage:=fbMessage) THEN
        fbMessage.GetTopic(pTopic:=ADR(sTopicRcv), nTopicSize:=SIZEOF(sTopicRcv) );
        fbMessage.GetPayload(pPayload:=ADR(sPayloadRcv), nPayloadSize:=SIZEOF(sPayloadRcv), bSetNullTermination:=FALSE);
    END_IF
END_IF

IF fbMqttClient.bConnected THEN
    IF NOT bSubscribed THEN
        bSubscribed := fbMqttClient.Subscribe(sTopic:=sTopicSub, eQoS:=TcIotMqttQos.AtMostOnceDelivery);
        IF fbMqttClient.bError THEN
            // add your error logging here
            hrErrorOccurred := fbMqttClient.hrErrorCode;
        END_IF
    END_IF
    fbTimer(IN:=TRUE);
    IF fbTimer.Q THEN // publish new payload every second
        fbTimer(IN:=FALSE);
        i := i + 1;
        sPayloadPub := CONCAT('MyMessage', TO_STRING(i));
        fbMqttClient.Publish(   sTopic:= sTopicPub,
                                pPayload:= ADR(sPayloadPub), nPayloadSize:= LEN2(ADR(sPayloadPub))+1,
                                eQoS:= TcIotMqttQos.AtMostOnceDelivery, bRetain:= FALSE, bQueue:= FALSE );
        IF fbMqttClient.bError THEN
            // add your error logging here
            hrErrorOccurred := fbMqttClient.hrErrorCode;
        END_IF
    END_IF
END_IF
```

Figure 3.24. MQTT code in Visual studio integrated with TwinCAT.

If the subscribing is not available, the code will automatically change to the publish mode and it is always waiting for messages to publish to a topic. By using an on-delay timer and triggering it, the messages for publish generates every one second. Two variables required for controlling the program sequence. These variables declare the topic and payload of the topic in the code and the messages sent and received every one second. Variable “bConnect” ensure the user that a connection from code to the broker is conducted and maintained.

- *MQTT client function block in Beckhoff TwinCAT*

Beckhoff used “TF6701 TC3 IoT Communication” library, which is basically a function block that enables TwinCAT software for connecting to MQTT. In mentioned library, there is a function block which enables connection with MQTT broker. “FB_IotMqttClient” is a function block responsible for accurately communication to MQTT broker. The method of “Execute()” should be cyclically called to ensure the send/receive communication is happening in the background. There are some other methods in this function block such as publish, subscribe, unsubscribe, ActivateExponentialBackoff and DeactivateExponentialBackoff that needs to be executed if needed depends on the case. The format and data type of the messages should be known to the receiver and sender parties, especially when binary or string data are sent. In this function block there are some inputs variables that need to define:

- sClientId: Type of this input is STRING (255), It consists of an individual client’s ID for the TwinCAT to the broker and if it does not specify, the software will generate it automatically during initialization.
- sHostName: Type of this input is STRING (255), It can be specified as IP address or name for the host and if it does not specify, the software will generate it automatically during initialization (the default value is local host .127.0.0.1)
- nHostPort: Data type of this input variable is UINT, host port should declare in this port. The default for this value is 1883.

- sTopicPrefix: This input is a prefix for the topic of subscribe/publish of the client and can be define internally. Data type of this input variable is STRING (255).
- nKeepAlive: is the watchdog time in seconds and it is monitored the communication between broker and client, Data type of this input variable is UINT.
- sUserName: for safety reason and ensuring the straight connection username and password can be define as STRING (255).
- sUserPassword: It is related password for the user identification, and it is optional. Data type of this input is STRING (255).
- stWill: This input is executed the structural block of “ST_IotMqttWill” which is another block in the “Tc3IotBase” library. The aim of using this input is to store a preconfigured message inside this variable for when the client disconnected from broker accidentally or maybe not regularly, then this message can be sent and stored in the broker as a “will topic”. This variable is also optional.
- StTLS: This input also connected to another data unit type, which is “ST_IotMqttTls”. Using this input is optional and it depends on the type of broker which is used. Some brokers offer a TLS-secured communication and for that kind of broker the security parameters such as authority certificate, client certificate and certificate revocation list can be added in this data unit and execute with the mention variable.
- IpMessageFIFO: Queueing of the messages are very important in MQTT communication, due to importance of real time monitoring and control it is necessary to record all the messages and events in the regular format with having time stamp and execution time of them. This variable is an instance of another function block “FB_IotMqttMessageQueue” which will explain further on. This variable used for storing the new incoming messages from the message queue when the call back is not implemented. In callback method functions are called in the code by a certain event and they are executed in a response to them. They are dependent on loops of client and without them callbacks are not able to trigger.

The output variables of the discussed function block can be explained as below:

- bError: A Boolean variable which becomes TRUE right after an error condition occurs in the communication between broker and client.
- hrErrorCode: This variable returns the error code from bError.
- eConnectionState: state of the communication between broker and client presents by this variable.
- bConnected: The value of this variable is TRUE when the connection between broker and client exists.

String variable formatting should be based on UTF-8 format which is a common type of string in MQTT protocol. UTF-8 is a character encoding format used for variable width. In this method first 128 characters of variable encoded to a single byte with the same value, so ASCII code can be encoded by UTF-8 and it is used in most programming languages. In order to receive this specific characters and data from all other languages and interprets them, Tc3_IotBase library includes character sets which is not restricted to typical character of string data type.

- *MQTT message formatting in Beckhoff TwinCAT*

Messages in “FB_IotMqttClient” function block are combined with two function blocks which are supervise on queueing and collecting the messages. “FB_IotMqttMessage” and “FB_IotMqttMessageQueue”. The incoming messages are identified with message queue then change them to understandable form for other function blocks. Available parameters such as payload size, topic and QoS are immediately provide as the output variables of this function block. By using some methods such as “GetTopic()”, “GetPayload()” and “CompareTopic()” the required data is executed from receiving messages. Another important parameter that need to considered is the size of the messages, although this size is depend on the hardware and memory of the PLC but it should not exceed from MB size even in high performance controllers. Tc3_IotBase library in this experiment has the possibility to define the maximum size of the messages by user. In this experiment the maximum size of messages is set on 100 KB. Applied method in this experiment can be explain as below:

- CompareTopic(): This method compare topics in the broker with the specified topic in messages and it returns TRUE when the desirable topic is find and match to the MQTT messages in each function block. Therefore, the required topic should be specified in the “sTopic” variable.
- GetPayload(): This method represents the content of the MQTT messages. Method type is Boolean and by using some input variables such as “pPayload”, “nPayloadSize” and “bSetNullTermination” this method is executable. Payload buffer, payload buffer maximum size and null termination payload is the application of mentioned variables.
- GetTopic(): Topic of MQTT messages is return by this method. Two variables are defined in this method, “pTopic” with a pointer to string data type. Memory address considered for the buffer that the topic should be copied there is specified in it. Moreover, “nTopicSize” related to maximum buffer size which is available, and it should declare in bytes.

The last function block which is applied in the MQTT communication of Beckhoff TwinCAT is “FB_IotMqttMessageQueue”. It offers the message queue capability for MQTT client and it can be declared as an instance of the client. Operation principle of this function block is base of FIFO. This processing method defines that first received data is the first data which should be send. During the execution of the code it is feasible to check the messages which and how many messages are collected from message queue. By using the “Dequeue()” method, it is possible to remove some messages from queue. In this case the oldest message will become the output first. Output variable “nQueuedMessage” can determine the amount of MQTT messages which are currently stand in the queue. Another input variable “bOverwriteOldestEntry” which is available as a property may be used to determine if new messages can overwrite to oldest messages in case of queue become full. If this variable is set to TRUE, oldest messages will be lost otherwise, just latest message will be dismissed. In case of queue size, it is possible to define the maximum size of the queue and allow the program to overwrite messages and loose oldest messages if the size of queue exceeded from specified value. By default, this value is set to 1000 messages. Messages size is restricted to 1000 KB, but for special applications it can be extended from parameter list menu.

4 Results and Discussions

By considering steps in previous sections and preparing the code in both Arduino and TwinCAT, the first experiment was conducted. In Arduino code it is important to determine some libraries such as “PubSubClient.h”, “WIFININA.h” and “ArduinoMqttClient.h”. Also, it is necessary to define the type MQTT client as “mqtt(net)” in Arduino code. Since the mosquito broker is running in the computer with the IP address of “192.168.0.101”, it is also required to specify this address as a MQTT server address for Arduino. As Arduino is the publisher in this condition and it should send the sensor data to the broker, publish top is declared in the code. Other parts of the code are mainly related to wifi connection and diagnostic and finally the payload which means how the message should transmit to the broker, In this experiment type of message shown as “{"Force": the value}”. After executing the code and download it on Arduino it will start to establish a connection with the message broker installed and run in the local host computer. By completing the requirement from Arduino then turn in to TwinCAT configurations. The important part in TwinCAT is defining the software as a simulator and use it as it works like PLC hardware. For doing that, it is necessary to change the CPU cores configuration. This can be done from “Real-Time” menu from solution explorer inside the software. It is important to have knowledge about the number of CPU cores. In setting tab, in section of “available cores” it is possible to set the cores for windows and simulation. By default, TwinCAT considered all the cores as shared CPU. This means that all the cores are shared between Windows and TwinCAT. This will cause to some severe errors. For solving this issue, the one core should be considered as shared and other cores (in this experiment three cores) as isolated cores. Then the TwinCAT can freely use these cores for simulating the code. By configuring the CPU cores and running the code in TwinCAT, it shows the real-time messages which is sent from Arduino inside the MQTT communication window. The value of Boolean variables as well as string variables are shown in this running experiment with their real-time values. Some of them were able to change online but some of them need to stop the simulation and then changing them. Figure 4.1 shows the output result of the integration in TwinCAT.

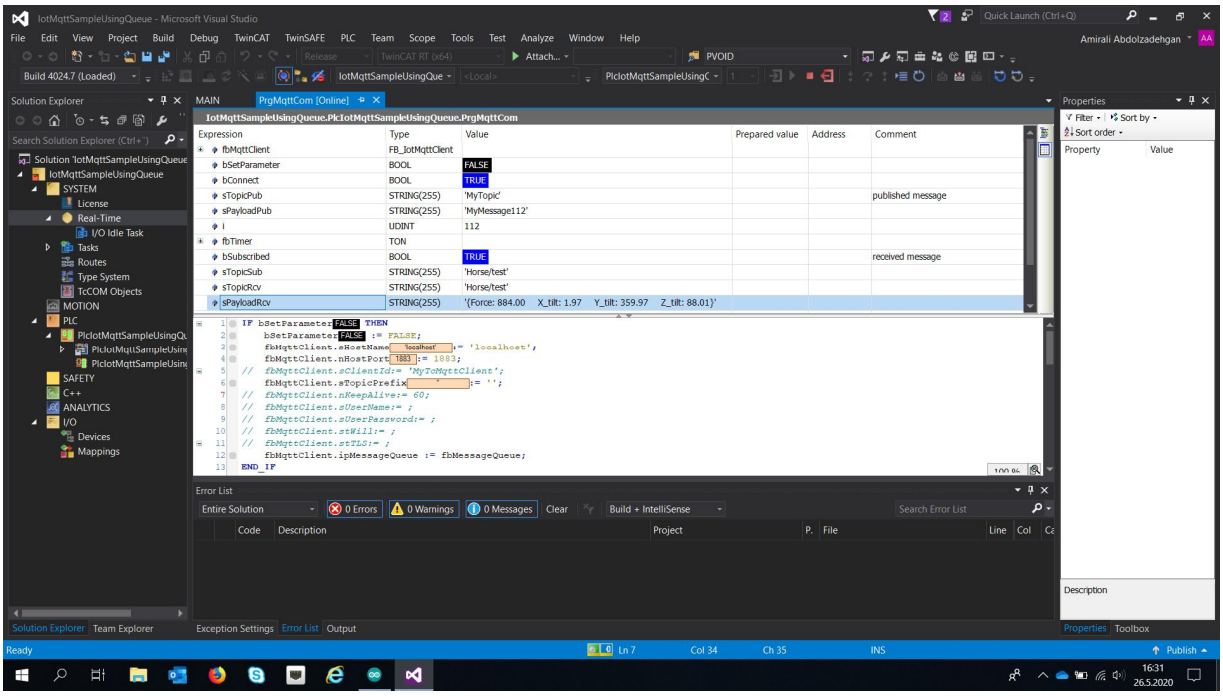

Figure 4.1. MQTT communication result in with TwinCAT.

This result shows that, the connection between TwinCAT and MQTT broker is conducted very simple and by considering some important consideration before running the code, it can guarantee the performance of the connection while the reliability and response time of the components in this network is reasonable.

5 Conclusion

Designing and integrating of sensors into a horse simulator includes different challenges. Considering various parameters such as acceleration, vibration and rider's safety makes the challenges even more complicated. In this study, sensor integration for measuring the saddle pressure, calf pressure and angle inclination was conducted. Wireless communication over MQTT protocol is considered as one of the most up to date and high-performance methods of IoT. It is based on publish/subscribe connection pattern and mainly used for machine-to-machine connection. Applying this protocol for networks which have delay experience is highly recommended. It contains high precision message queening approach, which is required to establish an appropriate data transfer. The objective of this study was investigating and having the experiment with this type of communication beside designing and integrating other sensor and automation devices in a horse simulator. MQTT is a simple messaging protocol which presents resource-structure network client via easy method for distributing telemetry. For this purpose, three separate experiments were carried out with different kind of platforms. The experiments include MQTT communication over internet by using a third-party message broker website, MQTT communication without internet by using third party software and wireless connection and finally MQTT communication without internet by using the Beckhoff TwinCAT software. The response time and accuracy of each method were studied along with the performance of each of them. While using internet is always convenient and it provides very high accessibility for the users and developers, the lack of proper protection and falling back from attackers may increase the risk of corruption. On the other hand the internet may not be available in some occasions and maybe some accidental operations may disconnect it.

From all the above-mentioned experiments, the following results were obtained which can be considered in further studies and future works with wireless networks and MQTT technologies:

- It is possible to carry out reliable and accurate data acquisition and use it for analysis and getting a good result from them. This method used a low-cost impact solution and available hardware with open source software for implementing other future projects or extending the current study.

- System integration in this study needs background in software development and programming skills as well as knowledge of sensors and their specifications. But lack of this experience will not affect too much. Since there are many available references and experts which shared their experience.
- The lightweight programming code allows to do a complex sequence with a few lines code. This communication is energy efficient, due to data packet minimization it can save battery power by using low amount of energy for CPU processing.
- Another experiment in case of disconnecting the stream shows that it is possible to continue the data transmitting from where it was disconnected. Or transmitting the data to other clients in this situation can be carry out.

And finally, to sum up all the findings and based on the results obtained from diverse platforms of experiments, using MQTT and wireless communication to sensors and controllers as the client and integrating all their messages into a broker, whether as a local host or maybe a web based broker, could simultaneously transmit messages and process parameters as a valuable and high-efficient communication method. This solution could develop the system’s availability to the acceptable level of standards which can be directed to the most of automation and IoT projects. This protocol is a great choice for lightweight data transmitting systems and definitely become very famous in the near future.

6 References

Amirat, Y., Francois, C., Fried, G., Pontnau, J. and Dafaoui, M., 1996. Design and control of a new six dof parallel robot: application to equestrian gait simulation. Mechatronics, 6(2), pp.227-239.

Byeon, Y.H. and Kwak, K.C., 2013, October. Analysis of domestic and international development trend for horse riding simulator. In 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013) (pp. 1258-1260). IEEE.

Chen, G.L., Wang, S., Kawata, K., Shinomiya, Y., Ozawa, T., Ishida, K., Kimura, T. and Tsuchiya, T., 2002, November. Biofeedback control of horseback riding simulator. In Proceedings. International Conference on Machine Learning and Cybernetics (Vol. 4, pp. 1905-1908). IEEE.

Dachyar, M., Zagloel, T.Y.M. and Saragih, L.R., 2019. Knowledge growth and development: internet of things (IoT) research, 2006–2018. Heliyon, 5(8), p.e02264.

Das, B. and Wang, Y., 2004. Isometric pull-push strengths in workspace: 1. Strength profiles. International Journal of Occupational Safety and Ergonomics, 10(1), pp.43-58.

Eskola, R. and Handroos, H., 2013. Novel horseback riding simulator based on 6-DOF motion measurement, a motion base, and interactive control of gaits. Advanced Robotics, 27(16), pp.1249-1257.

Koenig, P. and Bekey, G.A., 1993, July. Generation and control of lateral gaits in a horse-rider simulation. In Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93) (Vol. 1, pp. 572-579). IEEE.

Lee, M.W. and Kwak, K.C., 2013, August. 3D Motion analysis of national rider athletes by riding types in horse simulator. In Third International Conference on Innovative Computing Technology (INTECH 2013) (pp. 12-16). IEEE.

Lee, W., So, B.R., Lee, Y. and Moon, C., 2018. A new robotic horseback-riding simulator for riding lessons and equine-assisted therapy. International Journal of Advanced Robotic Systems, 15(4), p.1729881418784433.

Li, S., Li, L., Tao, J., Chan, L.C., Graham, J., Luk, T.C. and Chu, P.K., 2011, August. An Application Study on the Simulation Horse in Control Ability Evaluation of Jockeys Riding. In 2011 International Conference on Future Computer Science and Education (pp. 273-274). IEEE.

MacKinnon, J.R., Noh, S., Lariviere, J., MacPhail, A., Allan, D.E. and Laliberte, D., 1995. A study of therapeutic effects of horseback riding for children with cerebral palsy. Physical & Occupational Therapy in Pediatrics, 15(1), pp.17-34.

Nakajima, R., Shinomiya, Y., Sekine, O., Wang, S., Ishida, K. and Kimura, T., 1999. Horseback riding therapy system using VR technology, and toward to its medical evaluation. Trans Hum Interface Soc, 1, pp.81-86.

Sato, Y., Oshida, Y., Ohsawa, I., Sato, J., Yamanouchi, K., Chikada, K., Ito, K., Kato, K. and Sakamoto, N., 1991. Value of exercise in insulin-dependent and non-insulin-dependent diabetes mellitus. Diabetes, pp.241-244.

Shinomiya, Y., Ozawa, T., Hosaka, Y., Wang, S., Ishida, K. and Kimura, T., 2003, July. Development and physical training evaluation of horseback riding therapeutic equipment. In Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003) (Vol. 2, pp. 1239-1243). IEEE.

ARDUINO MKR WIFI 1010. (n.d.) Retrieved from https://store.arduino.cc/mkr-wifi-1010

BECKHOFF INFORMATION SYSTEM TWINCAT 3. (n.d.) Retrieved from https://infosys.beckhoff.com/index_en.htm