DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

RADIO REMOTE CONTROL SYSTEM OPERATING IN RESPECT TO OPERATOR POSITION

by
İlker BİRYOL

March, 2018
İZMİR

RADIO REMOTE CONTROL SYSTEM OPERATING IN RESPECT TO OPERATOR POSITION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering, Electrical and Electronics Science Program

by
İlker BİRYOL

March, 2018
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**RADIO REMOTE CONTROL SYSTEM OPERATING IN RESPECT TO OPERATOR POSITION**" completed by **İLKER BİRYOL** under supervision **ASST. PROF. DR. YAVUZ ŞENOL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master Science.

Asst. Prof. Dr. Yavuz ŞENOL

Supervisor

	Prof. Dr.
Dr. Öğr. Üy. Özgür Tamer	Mehmet Engin
(Jury Member)	(Jury Member)

Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Asst. Prof. Dr. Yavuz ŞENOL for his continuous guidance, generous support and valuable advice and for being a pleasure to work with. I have gained a different perspective through this study.

Finally, this study would not have been possible without the patience and everlasting support of my family and my friends. Especially thanks to my precious brother İdris BİRYOL and my dear friend Murat Gökay ZARARSIZ for his technical support. I would also like to thank my parents.

This thesis is dedicated to my deceased mother who has always been there somewhere to lend their unending support and love.

İlker BİRYOL

RADIO REMOTE CONTROL SYSTEM OPERATING IN RESPECT TO OPERATOR POSITION

ABSTRACT

Radio remote control is the use of radio signals to remotely control a system. The effective use of crane manipulation is an important contributor to industrial productivity, production efficiency, and especially workplace safety. Cranes are usually controlled either by cabled control unit or radio remote control units. Radio remote control devices are generally preferred for system safety and application efficiency.

Generally, cranes are used to move very heavy materials. Because of these movements of parts within complicated working places, there is possibility to have unintentional accidents and injuries. With currently available hand-held control devices, the commands obtained from actuators are fixed and they always activate the same motors at the same directions. During the crane control, crane operators move together with the payload and therefore they do change their positions in respect to get the best convenient position. When the operators change their position, they must be aware of changing commands to get correct movement of crane trolley and bridge. This thesis has investigated the overhead crane control systems to provide crane trolley and bridge movements in a way that the operator moves the joystick in respect to his/her standing position.

For this purpose, a prototype crane and the radio remote control system have been designed. In this design, a 3-axis digital compass was used to determine the position of the crane operator in respect to a reference position. With this position information, the crane control joystick signals were arranged to provide motor turnings in respect to position of the operator. The test results have shown that the designed system operate correctly and successfully from 0° to 360° around the crane, except some small critical positions. Therefore, the designed system is user friendly and provides safe crane working place.

Keywords: Radio remote control, radio signals, electrical overhead crane, safe crane control, user friendly

KULLANICI POZİSYONUNA BAĞLI ÇALIŞAN UZAKTAN KUMANDA SİSTEMİ

ÖZ

Radyo kontrolü, bir sistemi uzaktan kontrol etmek için radyo sinyallerinin kullanılmasıdır. Vinç sistemleri endüstri üretimine, verimliliğine ve özellikle çalışma alanı güvenliğine önemli bir katkı sağlamaktadır. Vinçler genellikle kablolu ya da kablosuz kontrol sistemleri ile kontrol edilmektedir. Radyo sinyalli kontrol sistemleri genellikle sistem güvenliği ve iş verimliliği için tercih edilmektedir.

Vinçler genellikle çok ağır malzemelerin taşınması için kullanılırlar. Karmaşık çalışma alanlarında parçaların taşınması esnasında istemsiz kazalar ve yaralanmalar olabilmektedir. Günümüzde bulunan kontrol sistemlerinde bulunan kontrol birimleri her zaman aynı yönde aynı motoru hareket ettirecek şekilde tasarlanmaktadır. Vinç kontrolü esnasında vinç operatörleri taşınan parça ile beraber hareket ederler ve en iyi pozisyon almak üzere konumlarını değiştirirler. Operatörler pozisyonlarını değiştirdikleri zaman, vinç araba ve köprüsünün doğru hareketini sağlamak üzere komutların pozisyonlarının farkında olmaları gerekir.

Bu tez, gezer köprülü vinç sistemlerinde vinç arabasını ve köprü hareketlerini operatörün ayakta durma pozisyonuna bağlı olarak, kumanda kolunun hareket yönüne uygun olacak şekilde çalışmasını sağlamak üzere bir araştırma yapmıştır. Bu amaç için hem prototip vinç ve hem de uzaktan kumanda sistemi tasarlanmıştır. Bu tasarımda 3-eksenli dijital kumpas referans noktasına bağlı olarak operatörün pozisyon tespiti için kullanılmıştır. Elde edilen pozisyon bilgisi ile vinç kontrol birimi kumanda sinyalleri motorların operatör pozisyonuna bağlı olarak hareket edecek şekilde düzenlenmiştir. Test sonuçları, tasarlanan sistemin bazı küçük kritik pozisyonlar haricinde, vinç çevresinde 0 ° 'den 360 °' ye kadar doğru ve başarılı bir şekilde çalıştığını göstermiştir. Bu nedenle, tasarlanan sistem kullanıcı dostu olup çalışma alanını güvenli kılmaktadır.

Anahtar Kelimeler: Uzaktan kumanda kontrolü, radyo sinyalleri, elektrikli gezer köprülü vinç, emniyetli vinç kontrolü, kullanıcı dostu

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Electrical overhead travelling cranes play an integral role in the construction and the other industries. These machines are used to lift, move, lower and raise the objects. Cranes have become the most essential and important mechanism in civil engineering and the other industries. Cranes hoist handle the material with chains or wire ropes. They are supposed to make the operations safely in a complicated working place.

Cranes oscillate under heavy loads like a pendulum, double pendulum or may have even more complex oscillatory dynamics (Strip et al., 1989). Various control have been developed to reduce oscillatory response from both issued commands and outer disturbances (Peng et al., 2009).

Crane operators who are operating convenient oscillation suppression technology generate safer and more efficient crane motions than operators without such facility (Khalid et al., 2006).

While important developments have been made to improve the operational efficiency of cranes by controlling their dynamic oscillatory responses, relatively minor matter has been given to the way in which operators issue those commands (Sorensen et al., 2008).

Most operators control cranes with push-button controller, joysticks, or control levers. These interfaces are not intuitive to many operators because they must first establish a mental map from the actuation of a button/joystick/lever to the actual crane motion that will be generated (Peng et al., 2009). As an example, the operator has to know that pushing button will make the crane travel in the forward direction. During the crane operation through the workspace and changes position, operators must continuously update this mental map. Besides, the interface is used to specify the movement of the overhead trolley, not the load.

For this reason, the operator must also calculate for the difference between the commanded movement of the trolley, which can be several meters overhead, and the delayed sweep response of the load (Peng et al., 2009).

Figure 1.1 is a schematic diagram that shows the basic components of a crane control system. The operator commands the crane by using some form of interface unit. Examples include pushing buttons on control pendants, pushing levers on a control or using joystick to the controller. Figure 1.1 is a block diagram that shows the basic steps of crane control. The interface evaluates this interplay and controls the command signal to the controller.

Generally this signal is electrical, and could be in the form of analog AC, or digital 1/0 inputs. Then the resulting command signal is sent to the crane, where the motors are energized and the actuators are actuated to generate crane movement (Peng et al., 2009).

Prototype Crane Control

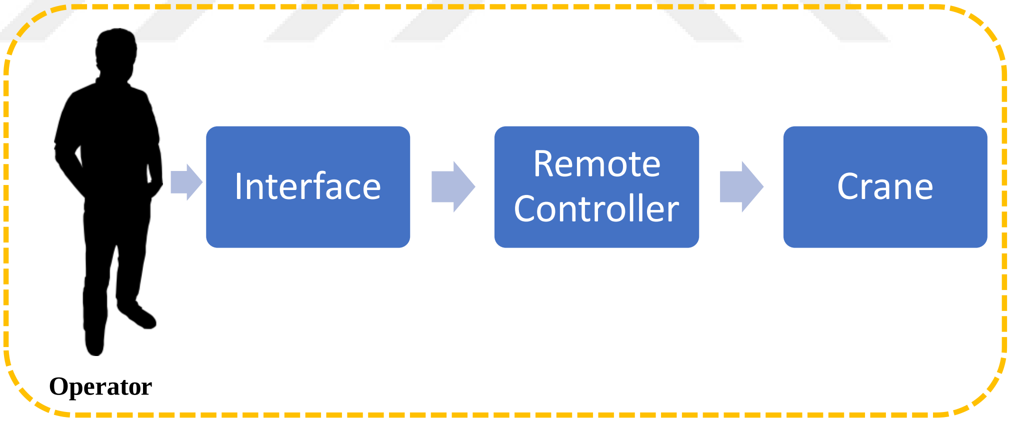


Figure 1.1 Crane operator, interface, and controller

The crane will act according to the physical dynamics of the crane system. It is interesting to note that there is not existing feedback loop. But, the motion of the crane is followed up by the operator; the operator then reacts accordingly by interacting with the interface, which can provide a close loop. The components of crane control, are defined by the larger control box, the operator, the interface, and the controller.

However, since we cannot design the operator, but the interface and the controller is included in the smaller control box, which include the components that are designable by design engineers (Peng et al., 2009).

With the termination of this research, a prototype crane was designed to operate the crane according to the user's position. And so, it is planned to create a safer working environment for operators.

CHAPTER TWO
LITERATURE REVIEW

2.1 History of Remote Control

Remote control devices are being used starting from 1970s. The main reason the start using remote control devices is to increase safety of operators and also efficiency of system. Initial radio systems use coded messages modulated to a carrier signal. The radio system consists of two parts; transmitter and receiver. The transmitter unit is used by the operator to control a remote system. The receiver used on the machine to transfer the given commands to related components. The transmitter and receiver units must uniquely be matched to provide safe communication area.

Current remote-control systems still have the same communication system, but with a more reliable an safe control functions. In addition to the radio signals infrared signals are used as a mean to provide line-of-sight starting position. Camera system for more complicated applications are used to have a wider view for the operator.

More sophisticated remote-control systems make it possible to control many machine systems from a signal transferring unit. This allows systems to act asynchronously or synchronously with each other. Generally, transmitter and receiver communicate with each other in two way communication, duplex. This allows the receiver to send many machine control information, which are positions, weights, faults, etc. information to the transmitter units. This information can be displayed to provide operator to aware of system functions.

2.2 Previous Works

In the field of Human Computer Interaction (HCI) research, there has been great interest around computer User Interfaces (UI), and how people interact with digital media. In many ways, the design of an intuitive crane-control interface parallels the design of UI, in that the aim of the interface are the same: easy learning, basic, and foreseeable. The designs of UI are often like the interfaces identified as interface-less, interface that leans heavily on the physical, non-figurative, and manage world; as opposed to relying on the user to understand the concept of abstraction that is commonly seen in Graphical User Interfaces (GUI) today (Lee et al., 2005).

Johnny Lee first gained wide-spread fame on the internet for his videos on hacking the Wiimote for low-cost UI's. The Wiimote is a video game controller that is standard with the video game console. He evaluates the infrared (IR) sensors in the Wiimote to track the motion of fingers, which are enlightened by an IR-LED array and reactive material fixed to the fingers. This opinion is used for a low-cost application of a multi-touch user interface. The same thought can be used to create an interactive whiteboard using an IR pen and a projector. Similarly, IR-LED's can be mounted to a pair of glasses to trace the motion of the head. The screen can be varied per head movement and is a low-cost option to high-cost Virtual Reality (VR) systems (Lee et al., 2008)

Lee's research follows a similar theme of using cameras, projectors, light pens, and light sensors to create foldable, movable, and interactive displays on surfaces. Interestingly, Raskar et al. gathering RF-ID tags and photosensors to project information in the application so that devices become self-describing and interactive with the operator (Raskar et al., 2004).

The Sixth Sense, created by Maes et al., is a wearable movement-based interface consisting of a camera, a projector, and colored markers fixed on the fingers. Views can be displayed on any stable surface by the projection devices. The operator can interact with the image using finger movements, which the camera can define owing to the colored markers (Maes et al., 2007).

In similar work, Jeffrey Han focused on applying low-cost solutions to largescale, multi-touch, back projection screens. Multi-touch screens are display surfaces whereby users can manage the displayed objects by touching the screen in a gestures. For example, turning an object can be accomplished by moving a finger in a round movement around the other fingers (Han et al., 2005).

The “Siftables", created by Merrill et al. and using for displaying mounted inside small plastic blocks that have different sensors. They have networking abilities that allow them to communicate with others also. (Merrill et al., 2007).

There is also similar investigation that focused on industrial implementation. Kazerooni et al. created Extenders, in which both a human and a robot apply important force to a load, with the robot exalting the human effort, much like that power managing amplifies the managing effort applied by an operator. A prototype arm-based extender was created for the application of equipment movements (Kazerooni et al., 1993).

The Magic Glove, is a notion that is like the glove based control discussed Kazerooni’s research. A major difference is that the magic glove operating pressure sensors to measure the force applied to a load. This data is sent as an RF signal to actuators that supply a proportionate amount of help to the operator (Kazerooni et al., 2004).

Many suppliers focus on ergonomic and intuitive equipment movement solutions. The iTrolley product range by Stanley Assembly Technologies is based on the concept of IAD's. The rigid-arm variant, senses operator purpose with a pressure based clutch. The wire-rope variant, uses angle sensors on the wire/rope to define user purpose when he/she pushes on the hanging load. (Peshkin et al., 2001)

Stanley Assembly Technologies also create the iLift product, which is a type of hoist. Gorbel, also creates air-balancers in a product line they call “Intelligent Lifting Devices". Air-balancers are a variant of cranes in that only the hoist raising or lowering works can be motorized. Lateral motions are provided by the operator’s push button hand held device to payload directly, as the trolley is free wheeled on the overhead tracks or railways. With this idea, the motion of the load is intuitive as there is no button pushing that is turning into movement. The operator raises or lowers the control grip to hoist or lower the load. Loads are typically in the order of a few hundred kilograms, and applications are often in small areas where operators are required to lift loads repetitively. Air-balancers also have a float-mode feature, where the motors exert just enough lifting force to set the weight of the payload. The operator can apply lesser amounts of additional force in the vertical direction so that payloads can be precisely orientated and positioned. (Peshkin et al., 2003)

For a human operator, rigid body behavior is much easier to predict than oscillatory behavior. The result of an intuitive interface is that it reduces the manual dexterity required for safe and efficient operation.

Wand control is a retro-reflective ball mounted to the end of a hand-held pole. A machine vision system is used to determine the position of the wand in real-time. The position of the wand is then used as the command signal to drive the crane (Peng et al., 2009).

Glove control is monotonically black, which is contrasted with a circular retro-reactive marker attached to the top-side of the glove. A machine vision system is used to determine the position of the glove in real-time. The position of the glove is then used as the command signal to drive the crane (Peng et al., 2009).

Radio-Frequency (RF) based control is Real-Time-Location- System (RTLS) based on Radio-Frequency Identification (RFID) technology is used to trace the position of a small tag held in the operator's hand. The position of the tag is then used as the command signal to operate the crane (Peng et al., 2009).

This thesis presents RF Based control in respect to operator position with Triple Axis Magnetometer. This interface allows operators to drive a crane simply by moving a remote controller joystics in 3-D space. This type of operator-motion interface addresses the issue of intuitiveness because it easy to learn, as motion with the operator remote controller to convey an aimed motion is familiar to most people (Peng et al., 2009). Additionally, this interface is simple, as it does not have a complex mental map between the interface and the crane response. This is virtually a seamless interface-less interface, as the mental map for this interface is simply a direct feed-through.

2.3 Principles of RF Based Interfaces

RF-based crane control relies on the use of RF sensors. The fundamental role of these interfaces is to acquire the position of the control device, which is then used as the command signal to drive the crane.

The goal of this section is to first present the hardware used for the experimental setup. Then, to comprehensively describe the hardware, software, and algorithms of the machine vision module. The section concludes with a description of the interface for the RF-based crane control.

2.3.1 Crane Control Interfaces

The impact of the crane users has a big effect on the safety of the working area and productivity of the factory or plant. One of the significant thing that makes crane system control more difficult is the load tends to sweep like a pendulum, a double pendulum or even bouncing dynamics (Kivila et al., 2013).

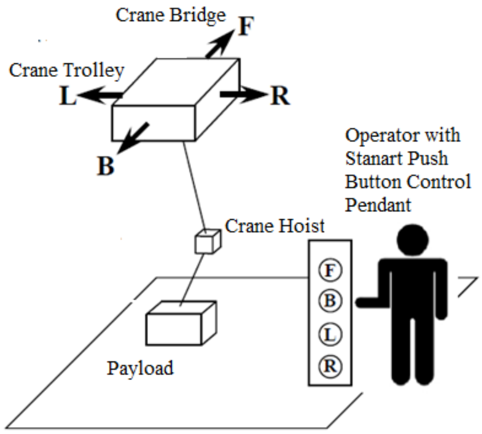


Figure 2.1 Crane control using a push-button pendent (Kivila et al., 2013).

Controlling a crane is frequently very difficult for crane users because of the slow response of the heavy structures and the lightly-damped load sweep. Falsification tasks are made even more difficult when the interface between the crane user and crane is un-intuitive (Kivila et al., 2013).

Additionally, the complicated dynamics of the crane, operators must master the control interface. Figure 2.5 shows pendent control of a typical overhead crane. The load is attached to the hook, forming a double pendulum system. The operator must transfer the desired payload path into a sequence of button presses that produces desired crane movements (Kivila et al., 2013).

For example, if the crane user wants to drive the crane through a working place, then the desired path must be designed into a sequence of scenes where the forward (F), backward (B), left (L), and right (R) buttons are pushed at the right times. Additionally, the crane user can move through the working space to tracking the load, which is usually causes the cranes users to change the path they are facing with. These kin of shifts cause the forward button on the standard control pendants to replace the crane in a direction that is not in the operator’s forward direction. It has been shown that requiring such analytic problem solving during crane operation consequences in more mistakes (Reason et al., 1990).

Efficiency effect of the varied interfaces have been detected by conducting tests where crane users used a dynamically reparations crane through a series of point-to-point paths. The dynamic reparations let the crane's load to reply to commanded movement with minimum cable oscillation, in any case of which crane user interface was used. The time to complete every motion was booked for each respective path and interface apparatus (Sorensen et al., 2007).

The results of previous trials showed that crane users derived the dynamically compensated crane more adequately using the visual interface than when using either the pendent or joystick. Even so, only a limited amount of paths has been tried. These occurred of different straight-line motion in different route that did not contain pulling of the load (Sorensen et al., 2007).

CHAPTER THREE
MATERIALS AND METHODS

This chapter introduces which material is used in this experiment and the thesis and also, how the design is implemented on the prototype device. Additionally, what is the algorithm behind the interface between crane prototype and radio remote controller.

3.1 Design of Study

Structure of the prototype crane is designed at “Autodesk Fusion 360”. The structure includes complete crane system and its construction consisting of rails, columns, trolley, hoist unit, wheels and other components. Figure 3.1 shows the structure design of rails and columns.

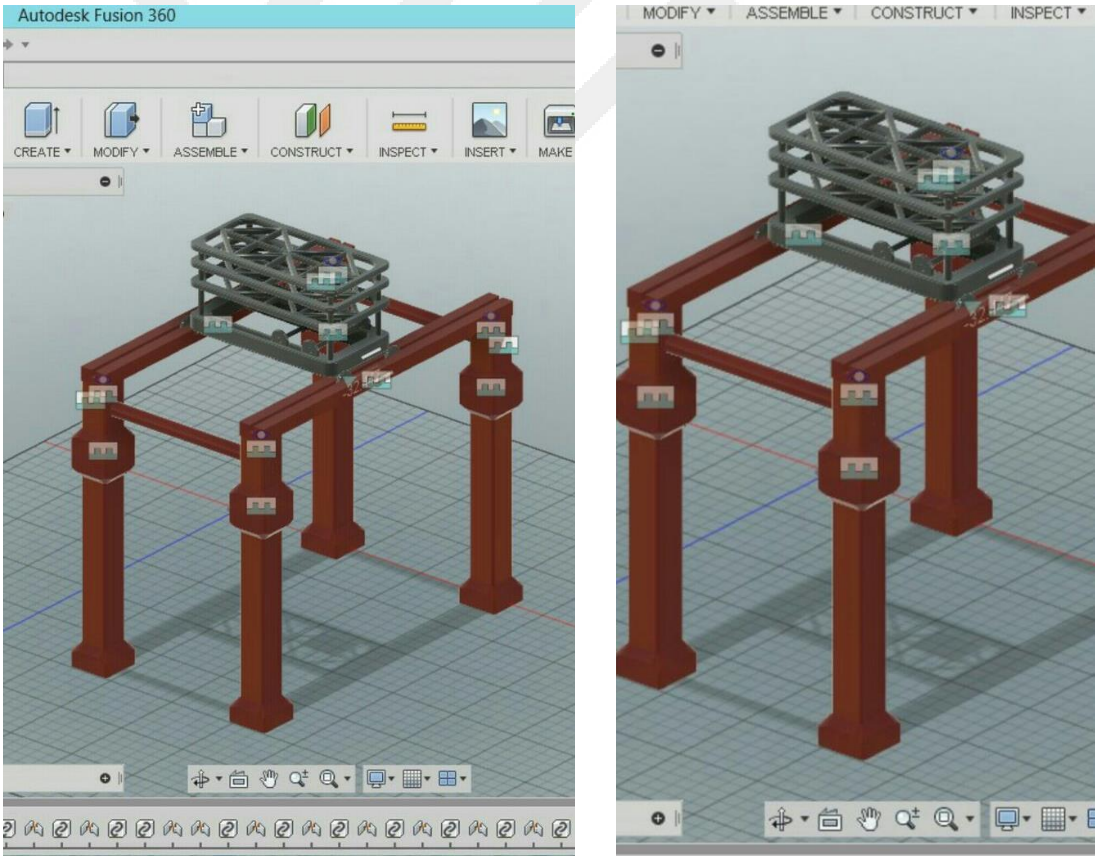


Figure 3.1 Autodesk fusion 360 columns and rails structure design

After the design completion, the next process is the to get all components of the prototype crane from a 3-D printer. For this printing job PRUSA I3 3D PRINTER used. In these type 3D printers filament material is used as a base material. This printer has approximately 19x20x17 cm working dimensions, and have 20A power supply.

3.2 Crane Hardware

Bridge cranes are lifting systems include four parts: a hoist cable and hook, a trolley, a bridge, and a runway. The payload is attached to the hook, which is suspended from the trolley by the hoist cable. The trolley is the lifting part and moves on (and parallel to) a beam called the bridge. The bridge moves on (and parallel to) a fixed runway (Sorensen et al., 2005).

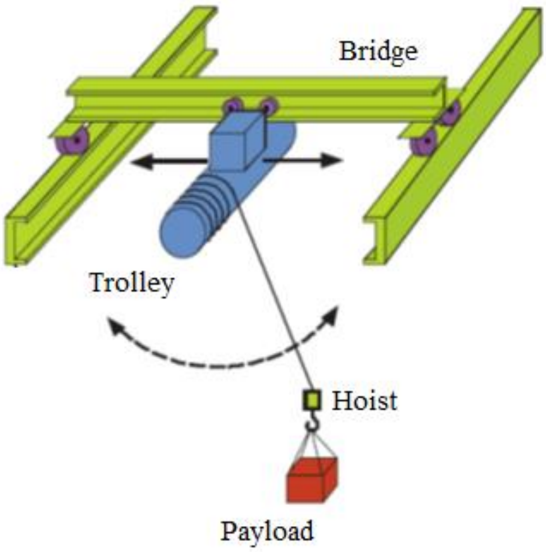


Figure 3.2 Model of a bridge crane (Sorensen et al., 2005)

The bridge and runway span a 30 cm by 660 cm^2 area. The hoist system is mounted to the bridge, which is suspended above the ground at a height of 18.5 centimeters.

A schematic of the hardware components of the crane are shown in Figure 3.2. The crane uses four 12-Volt DC motors to drive the bridge and one to drive the trolley. And one Arduino Mega used to control the bridge and trolley motors.

Figure 3.3 Prototype crane (Personal archive, 2017)

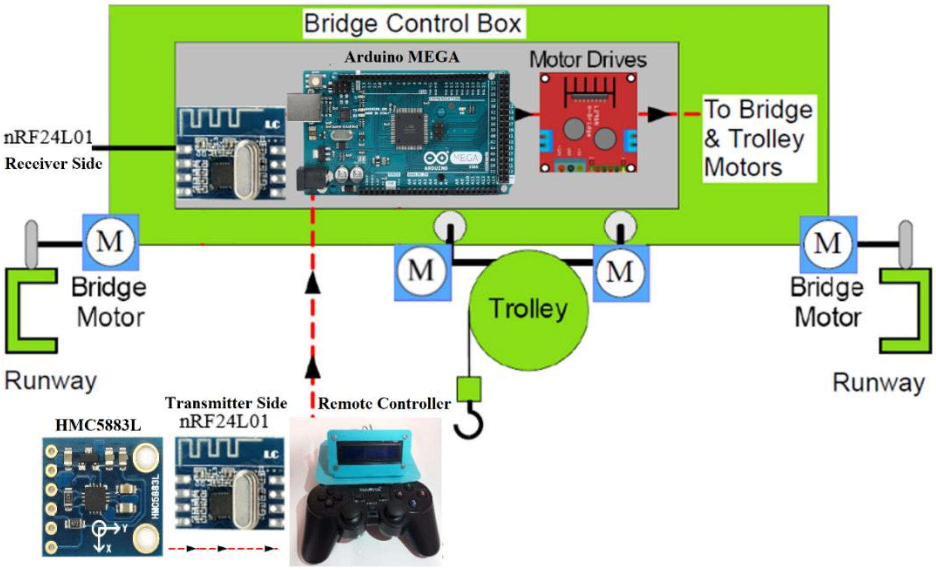


Figure 3.4 Hardware configuration of prototype crane

A schematic of the hardware components of the prototype crane are shown in Figure 3.3. Under advanced-control operation the Arduino Unit intercepts the signals from the control pendent to the motor drives.

Position measurements of the trolley location are provided by one HMC5883 Triple Axis Magnetometer along the bridge and trolley axes. The magnetometer sensor giving feedbacks to receiver unit. A HMC5883 sensor determine the position related with control pendant. The Arduino Unit is placed the bridge control box. That Arduino Unit connected to L298N motor driver unit to drive motors related with HMC5883 magnetometer. Communication is established via a nRF24L01 network module that uses RF protocol.

3.3 Configuration used in Experimental Studies

A remote controller is a part of our control system which is used to operate the device wirelessly from a distance. The remote controller which is used our experiment consist of LCD panel, HMC5883L 3-axis digital compass and nRF24L01 wi-fi module, and mounted on a gaming console which is shown Figure 3.3 and Figure 3.4.

3.3.1 Arduino

The experimental study is realized based on Arduino, for this study 2 Arduino UNO and 1 Arduino MEGA microcontrollers were used. Two of the Arduino UNO microcontroller were used for the transmitter of the remote controller unit. One of these was dedicated for LCD panel control placed at the remote controller. The other Arduino UNO was used for handling joystick signals and the programming the transmitter. Whereas, the Arduino MEGA is used for receiver side of the experimental system for controlling the motors.

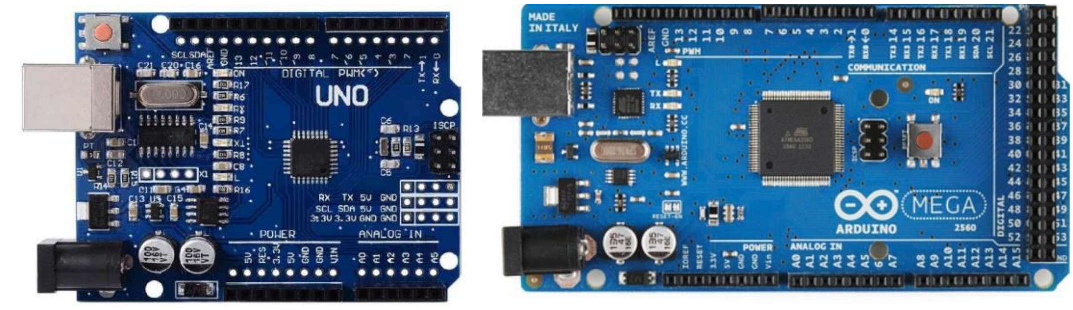


Figure 3.5 Arduino UNO and Arduino MEGA

3.3.2 L298N Duel Motor Driving Card

For driving the motors L298N Duel Motor Driving Card used. This card can drive motors which is working DC voltage. It has two channel and every channel can load 2A. It has also high temperature protection with internal cooler.

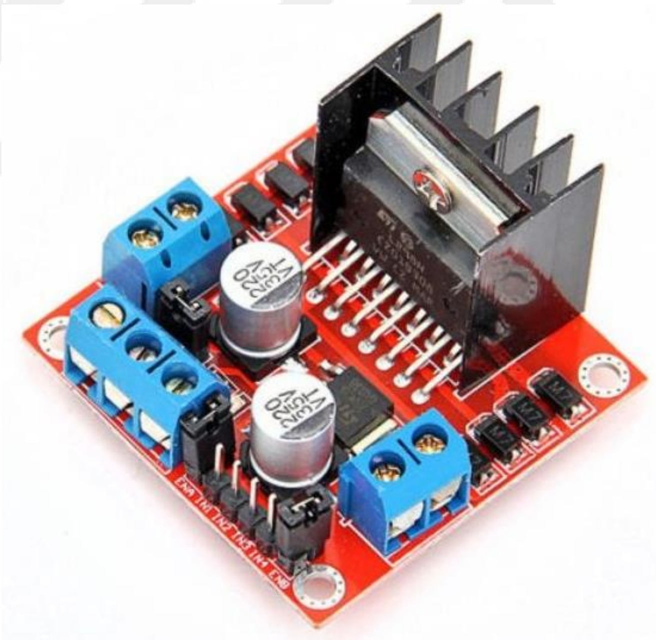


Figure 3.6 L298N dual motor driving card

This card allows to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC.

3.3.3 RPM Reducer DC Motor

This gearmotor consists of a high-power, 12 V brushed DC motor combined with a 9.68:1 metal spur gearbox. The gearmotor is cylindrical, with a diameter just under 25 mm, and the D-shaped output shaft is 4 mm in diameter and extends 12.5 mm from the face plate of the gearbox.

These cylindrical brushed DC gearmotors are available in a wide range of gear ratios and with five different motors (two power levels of 6V motors and three power levels of 12V motors). The gearmotors all have the same 25 mm diameter case and 4 mm diameter gearbox output shaft, so it is generally easy to swap one version for another if your design requirements change though the length of the gearbox tends to increase with the gear ratio.

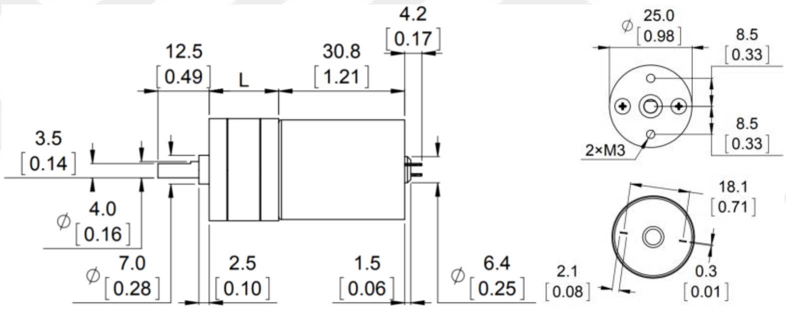


Figure 3.7 Dimension diagram of DC motor

Figure 3.8 1000 RPM reducer DC motor

3.3.4 NRF24L01 Wi-Fi Module

The nRF24L01 is a card which have 2.4GHz transceiver capacity with an embedded baseband protocol engine suitable for ultralow power wireless applications. The nRF24L01 is designed for process in the world-wide ISM frequency band at 2.400 - 2.4835GHz.

The nRF24L01 control and format through a Serial Peripheral Interface (SPI). The register map, which is reachable through the SPI, include all format registers in the nRF24L01 and is reachable in all operation modes of the card.

The embedded baseband protocol engine is based on packet communication and supports different types from manual process to advanced automatic protocol process. Internal firs in first outs provide a smooth data transfer between the radio front end and the system's microcontroller unit.

The radio front which mean transmitter side end uses Gaussian Frequency Shift Keying modulation. It has user adjustable parameters like frequency channel, output power and air data ratio. nRF24L01 module supports an air data rate of 250 kbps to 2Mbps.

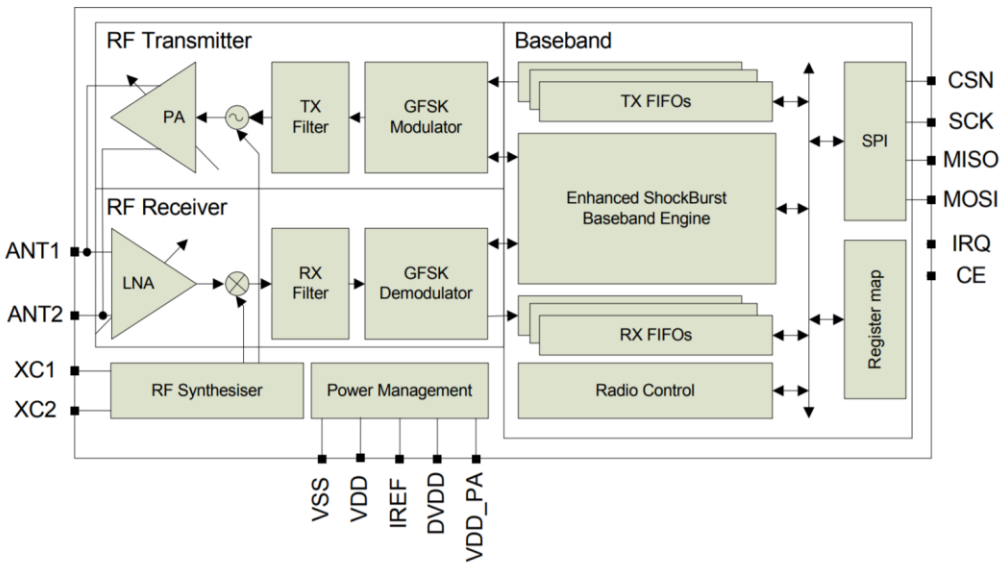


Figure 3.9 nRF24L01 block diagram

Figure 3.10 nRF24L01

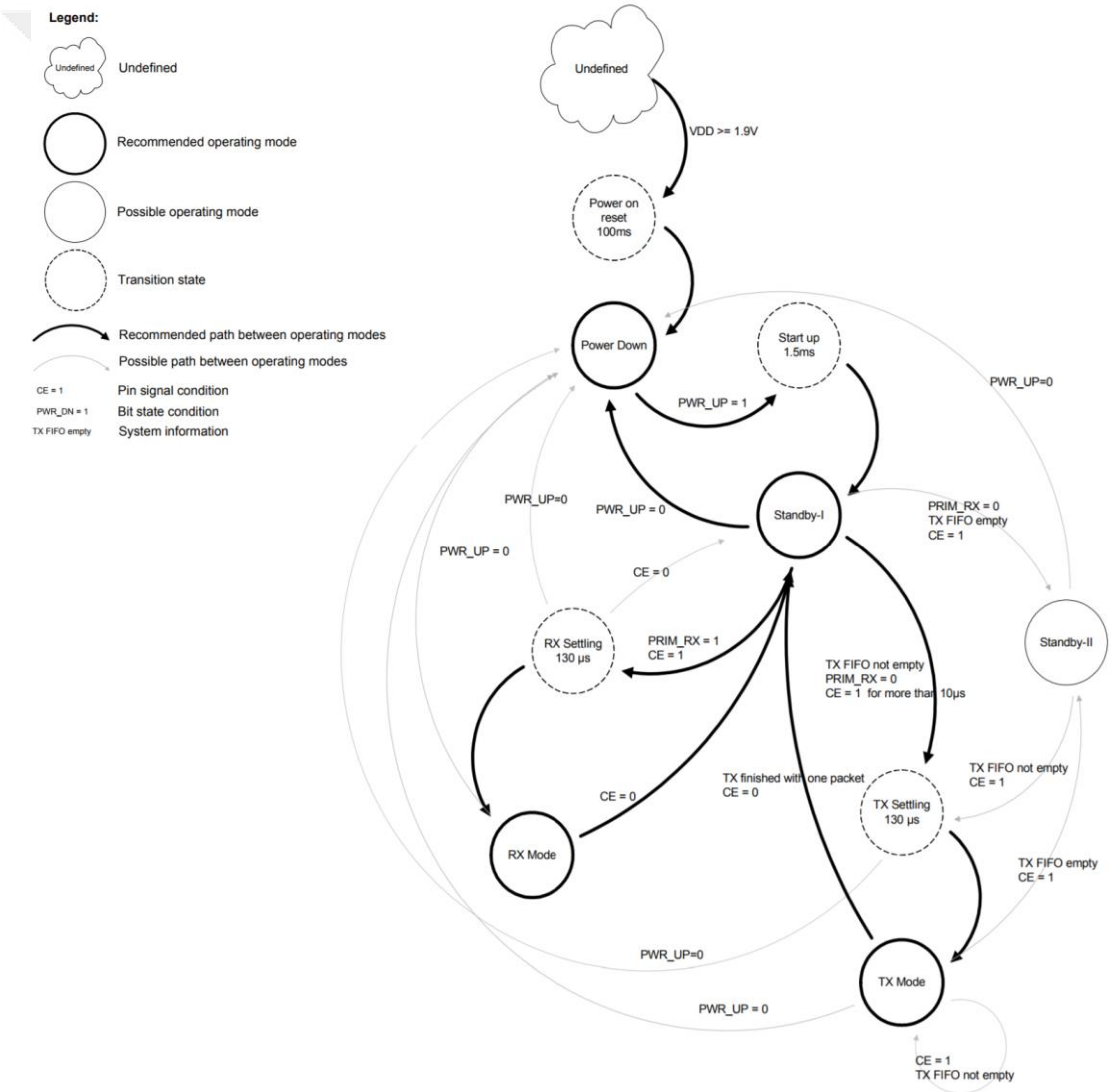


Figure 3.11 Radio control state diagram of nRF24L01

3.3.5 HMC5883L 3-Axis Digital Compass

HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. Magneto-resistive sensors plus, offset cancellation, and a 12-bit analog to digital converter enables 1° to 2° compass sensivity. The I^{2}C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC).

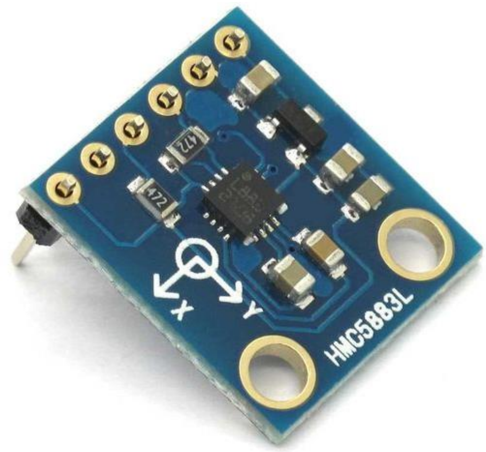


Figure 3.12 HMC5883L 3-axis digital compass

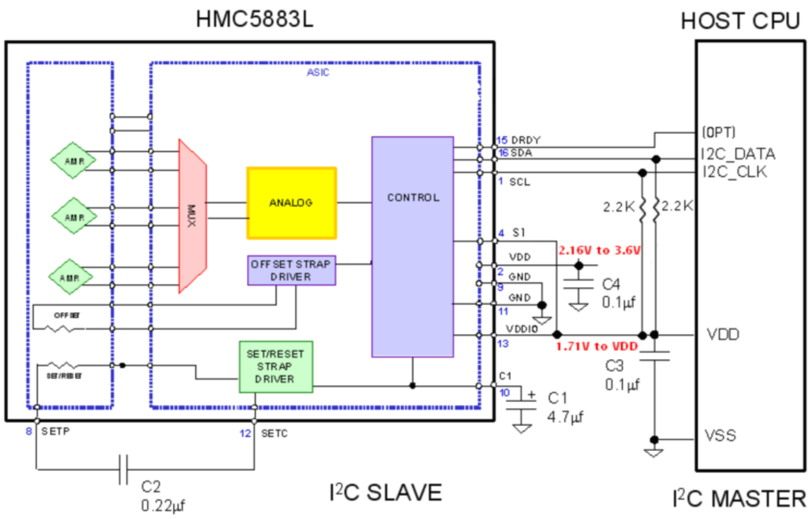


Figure 3.13 Internal schematic diagram of HMC5883L

The HMC5883L corresponds with a two-wire I^2C bus system as a slave card. The HMC5883L uses a basic protocol with the interface protocol defined by the I^2C bus specification. The data rate is at the standard mode 100kbps or 400kbps rates as determined in the I^2C bus specifications. The bus bit format is an 8-bit Data/Address send and a 1-bit acknowledge bit. The format of the data bytes (payload) shall be case sensitive ASCII characters or binary data to the HMC5883L slave, and binary data returned.

The HMC5883L Serial Clock (SCL) and Serial Data (SDA) lines require resistive pull-ups (Rp) between the master device usually a host microprocessor and the HMC5883L.

The serial clock and serial data lines in this bus specification may be connected to other devices. The bus can be a single master to multiple slaves, or it can be a multiple master format. All data transfers are started by the master device, which is liable for creating the clock signal, and the data transfers are 8-bit long.

Per the I^2C spec, all progression in the serial data line must consist when serial clock is low. This necessity leads to two matchless circumstances on the bus incorporated with the serial data transitions when serial clock is high. Master device pulling the serial data line low while the serial clock line is high defined the Start (S) condition, and the Stop (P) condition is when the serial data line is pulled high while the serial clock line is high. The I^2C protocol also let us for the “Restart” condition in which the master device issues a second begin condition without issuing a stop.

I^2C bus check can be applied with either hardware logic or in software. Characteristic hardware designs will discharge the serial data and serial clock lines as suitable to let the slave device to control these lines. In a software application, care must be taken to perform these tasks in code.

3.4 Working Logic of Prototype Crane

A prototype crane was designed to create operator based remote control system. Different electronic components are used which will be mentioned in this section.

Firstly, the construction of the crane was realized by Autodesk Fusion 360 and printed at Prusa I3 3D Printer. Printed parts shown in Figure 3.14.

Figure 3.14 Printed parts of prototype crane construction (Personal archive, 2017)

The printed parts were assembled together by use of a type of special glue. In this way the frame of the crane system was completed. The trolley of the crane was also printed with a 3D printer with different filaments.

After the completion of the construction, the electronic components were installed. Two 1000 RPM Reducer DC Motors were installed on the end carriages of overhead crane construction to actuate the bridge. Same motors were also used to actuate trolley of the prototype overhead crane. Finally, a 600 RPM DC motor was used to control hoist unit of trolley.

These motors were driven with L298N Duel Motor Driving Card. This card can drive two motors at the same time. The bridge and trolley motors were driven with one L298N. Two motors connected in parallel with drive bridge and the other two motors, which are also connected in parallel, used to drive the trolley. This connection diagram showed at Figure 3.15. The other 600 RPM motor driven by Arduino unit is used to provide up and down movements of the trolley hoist. Figure 3.16 shows additionally wi-fi module and the other motor connections at the actuator side.

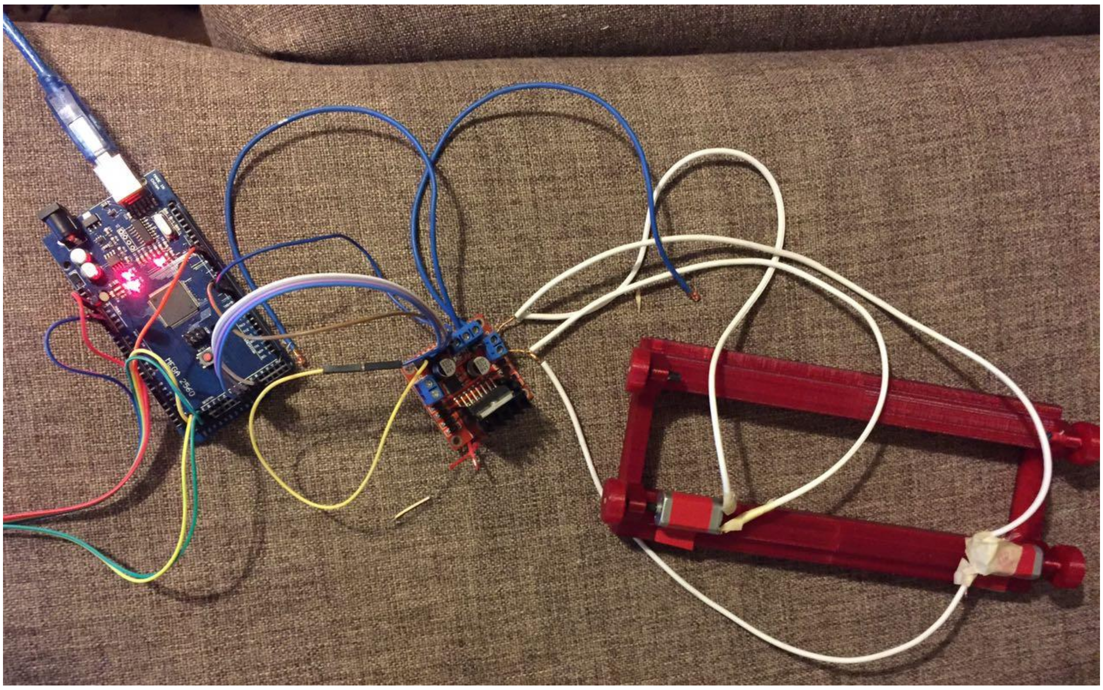


Figure 3.15 Bridge motors and L298N drive card connection (Personal archive, 2017)

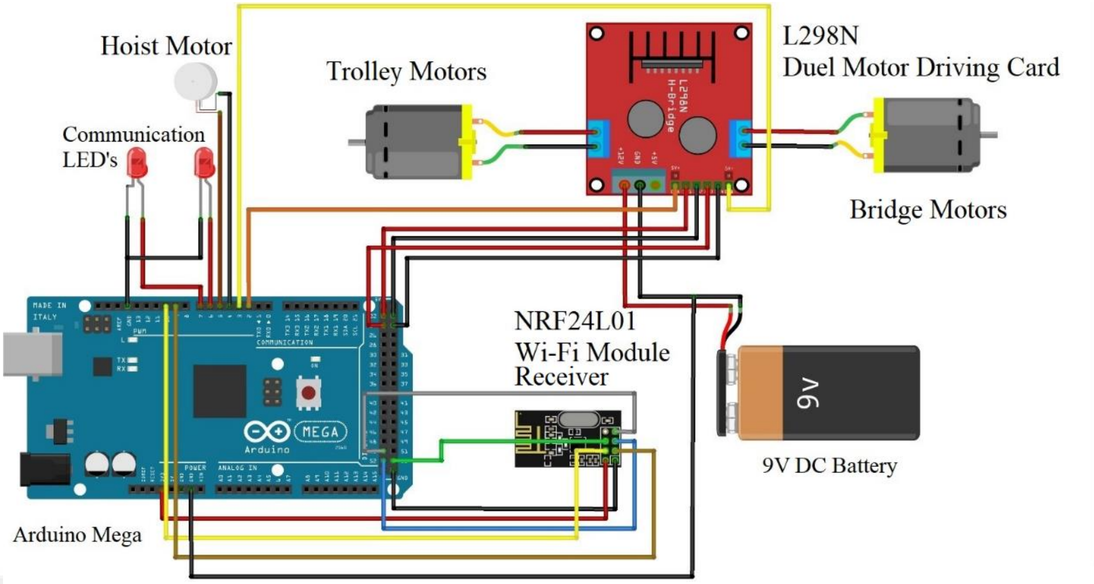


Figure 3.16 Schematic diagram of the receiving side of the prototype crane system

The prototype crane is controlled with a radio remote control and the commands from the operator, which are provided with the transmitter unit, has to be transferred to the receiver unit placed at the crane hoist electrical panel. The signal transmission between transmitter and the receiver is through RF based communication. For this purpose, NRF24L01 Wi-Fi Module was used. One of NRF24L01 Wi-Fi Module is used at the receiver and the other one is used in the transmitter of the radio remote control unit. The only difference is the written code which is defined at the Arduino Uno and Mega.

The turning direction of these motors changes related to position of the operator in respect to crane. The position of operator in respect to crane is decided by use of HMC5883L 3 Axis Digital Compass. This digital compass is placed within the transmitter part of the radio remote device. Figure 3.17 shows the connections between the transmitter, Arduino Uno, NRF24L01 Wi-Fi Module and HMC5883L 3 Axis Digital Compass.

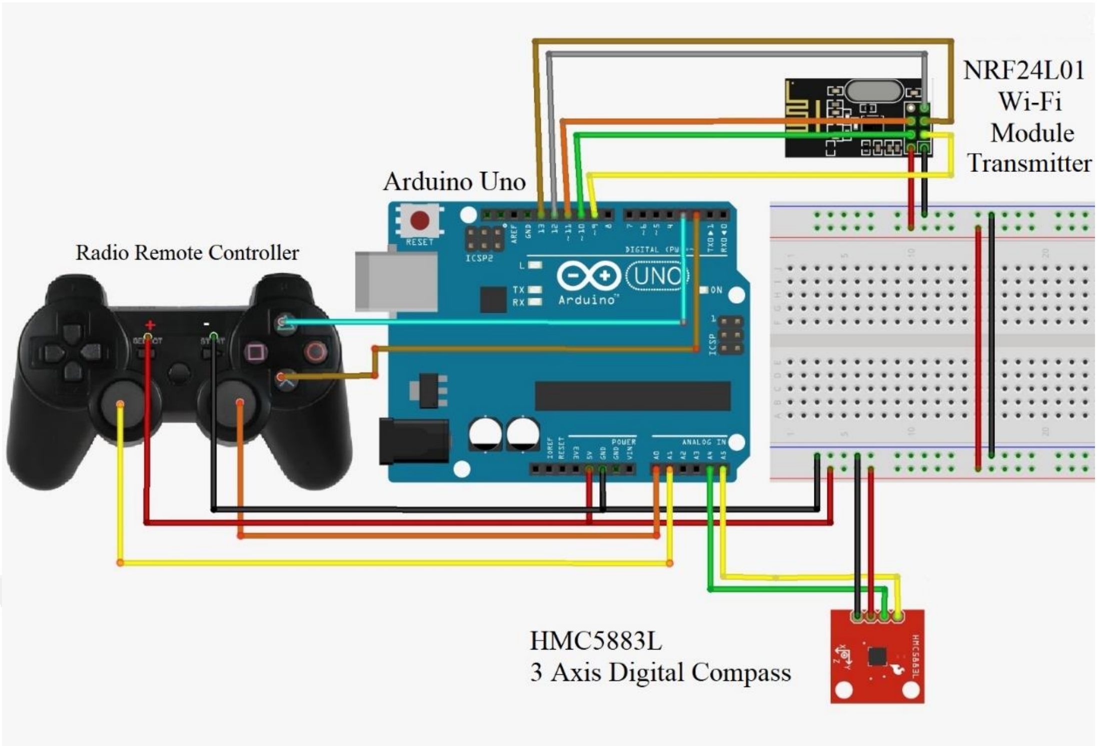


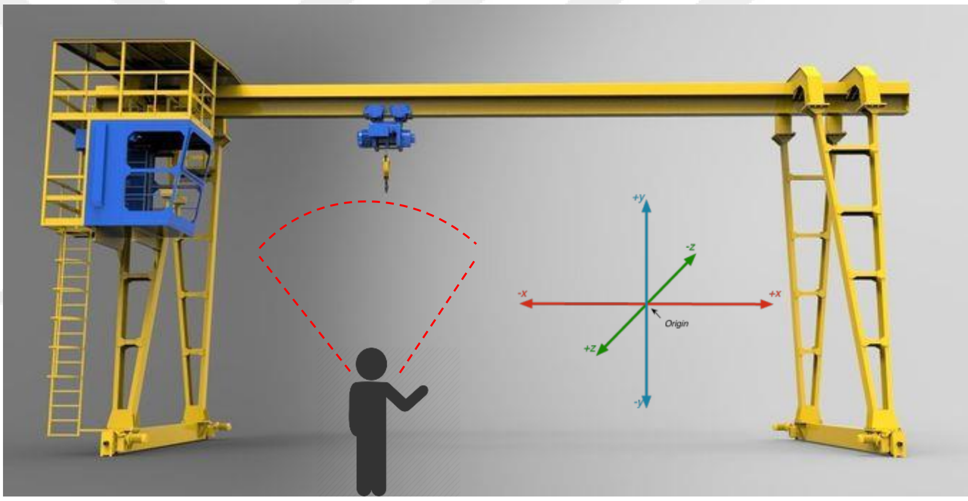
Figure 3.17 Schematic diagram of transmitting side of the prototype crane system

CHAPTER FOUR
RESULTS AND DISCUSSION

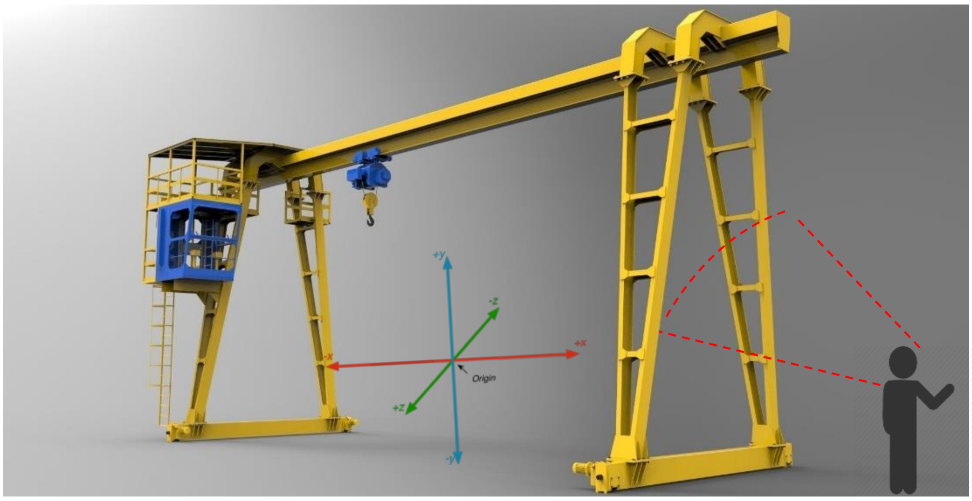
This chapter presents the research results of experimental study on radio remote control in respect to operator position related with the crane.

4.1 Performance of Experimental Study

All control units were programmed to provide related prototype crane movements. The crane operator provides all crane movements using radio remote control joystick. Figure 4.1 gives different crane movements in respect to operator's position.

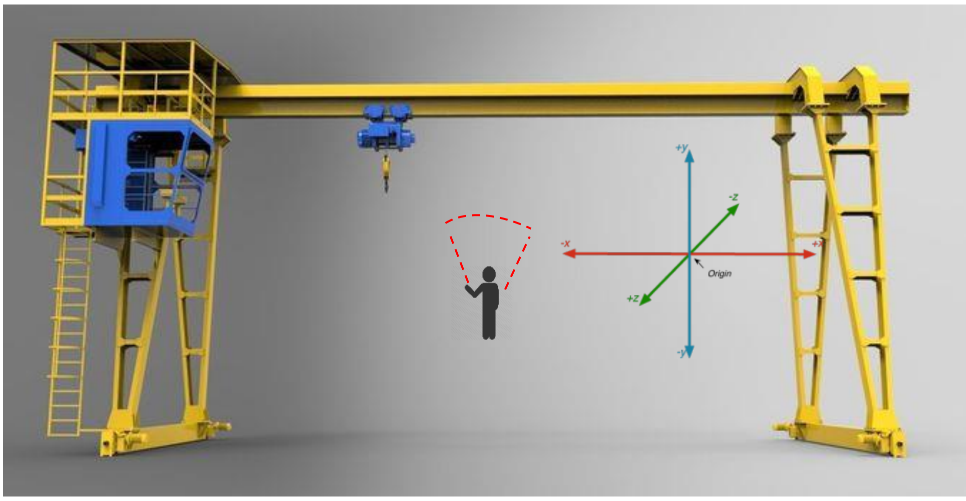


(a)



(b)

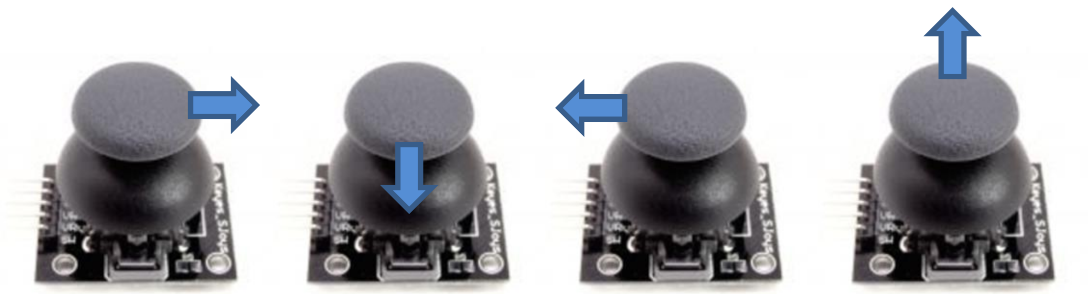
Figure 4.1 Operator positions related to gantry crane



(c)

(d)

Figure 4.1 continues



(a)- Right (b)-Back (c)-Left (d)-Forward

Figure 4.2 Joystick movement

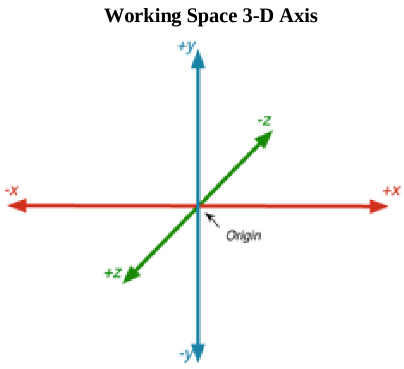


Figure 4.3 Working space of prototype crane

The idea is to fix the joystick commands even when the operator changes his/her position in respect to crane. This can be only provided by changing joystick commands when the operator changes his/her direction at certain angles. For example, looking from one direction to the hoist, trolley moves right with right movement of joystick actuated. When the operator changes his/her position and looks the hoist at opposite direction, normally trolley moves left when the operator moves the joystick to the right direction. However, here with correct calculation of operator position in respect to crane, this is prevented and the trolley moves right with right movement of the trolley joystick. With standard radio remote control units, crane bridge and trolley movement directions do not change in respect to the position of the operator.

If we accept Figure 4.1 as our working space, and the Figure 4.1 (a), (b), (c) and (d) shows different position of the operator in respect to crane. Figure 4.2 also shows different radio remote controller joystick movement directions. According to this information the crane's bridge and trolley movements are set related with the obtained crane operator positions and are described below.

If the operator stands at a position as given Figure 4.1 (a), moving the joystick to the right as in Figure 4.2(a) trolley motor accelerates at the +x axis. In this position if we move the joystick back as in Figure 4.2(b) the bridge motor accelerates at the +z axis, and moving the joystick at left as in Figure 4.2(c) trolley motor accelerates to -x axis. Finally, when we move the joystick forward as in Figure 4.2(d) the bridge motor accelerates in -z axis in this position of the operator.

When the operator changes his position to stand as given in Figure 4.1 (b), moving the joystick to the right as in Figure 4.2(a) bridge motor accelerates at the -z axis. In this position if we move the joystick back as in Figure 4.2(b) trolley motor accelerates at the +x axis. If we continue moving the joystick to the left as given in Figure 4.2(c), the bridge motor accelerates to +z axis. Finally, when we move the joystick forward as Figure 4.2(d) the trolley motor accelerates at -x axis in this position of the operator.

When the operator is in a position as given in Figure 4.1 (c), which is opposite to the direction given in Figure 4.1 (a), moving the joystick to the right as in Figure 4.2(a) the trolley motor accelerates at the -x axis. In this position if we move the joystick back as in Figure 4.2(b) the bridge motor accelerates at the -z axis, and continue moving the joystick to the left as in Figure 4.2(c) the trolley motor accelerates to the +x axis. Finally, when we turn the joystick forward as in Figure 4.2(d) the bridge motor accelerates to the +z axis in this position of the operator.

When the operator is in a position as given as in Figure 4.1 (d), which is opposite to the direction given in Figure 4.1 (b), moving the joystick to the right direction as in Figure 4.2(a) the bridge motor accelerates at the +z axis. In this position if we move the joystick back as given in Figure 4.2(b) the trolley motor accelerates at the -x axis. If we continue moving the joystick to left direction as in Figure 4.2(c) the bridge motor accelerates to -z axis. Finally, in the same position of the operator when we move the joystick forward direction as in Figure 4.2(d) the trolley motor accelerates to +x axis. Crane’s trolley and bridge movements controlled by the crane operator in respect his/her position are summarized at Table 4.1. In the designed radio remote control system the actual crane motor movement directions are calculated as soon as the operator moves the joystick. These movement direction calculation’s flow chart is given in Figure 4.4.

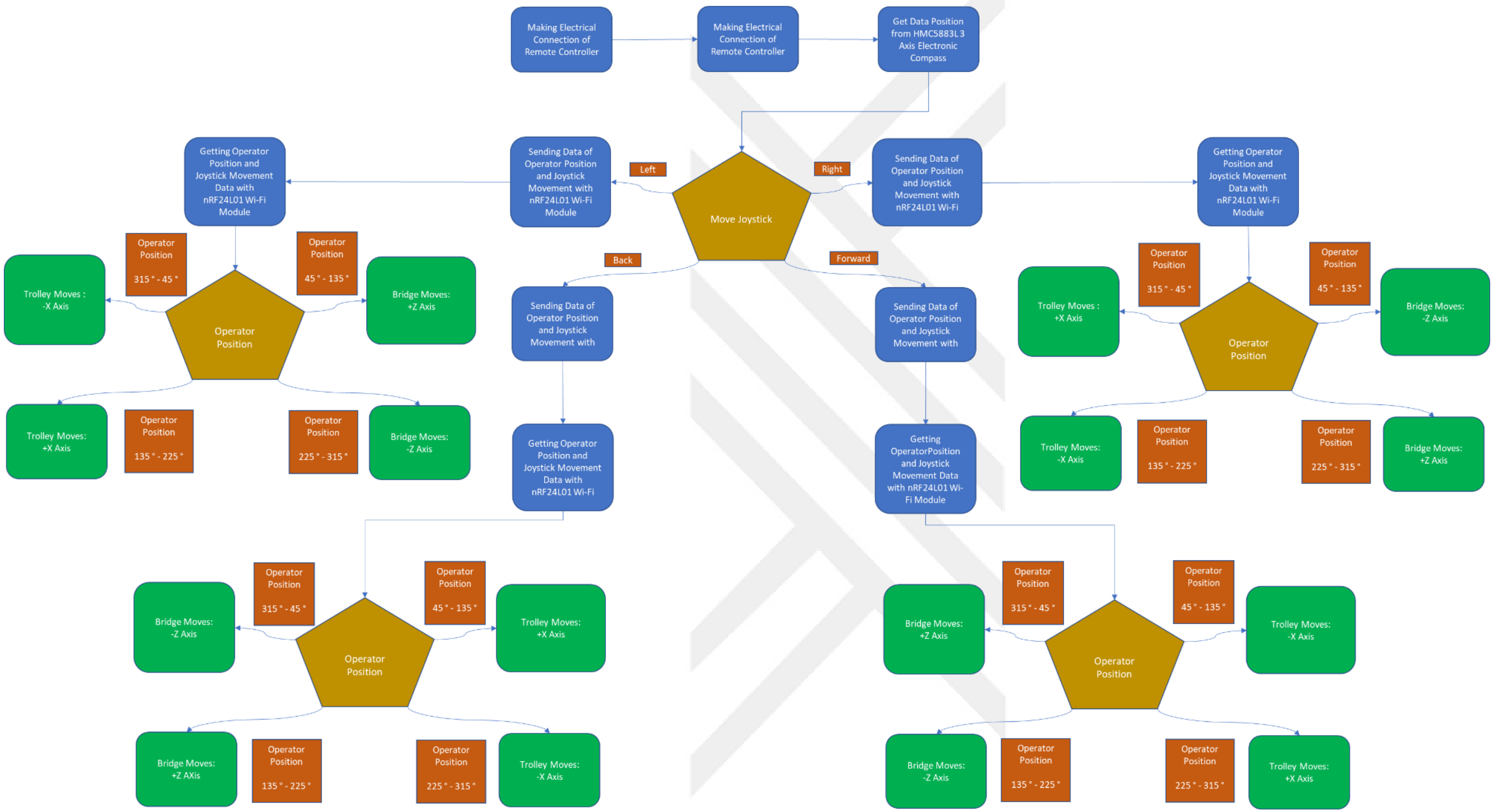


Figure 4.4 Prototype crane flow chart

An experimental setup was prepared to check the accuracy of the system. The relevant experimental unit consists of one stationary and one moving proposal table. Scaled paper with grade is prevented from sticking on the fixed table. In addition to this, the moving table is fixed with a gauge to keep the remote-control stationary. The relevant experimental setup is shown in Figure 4.5.

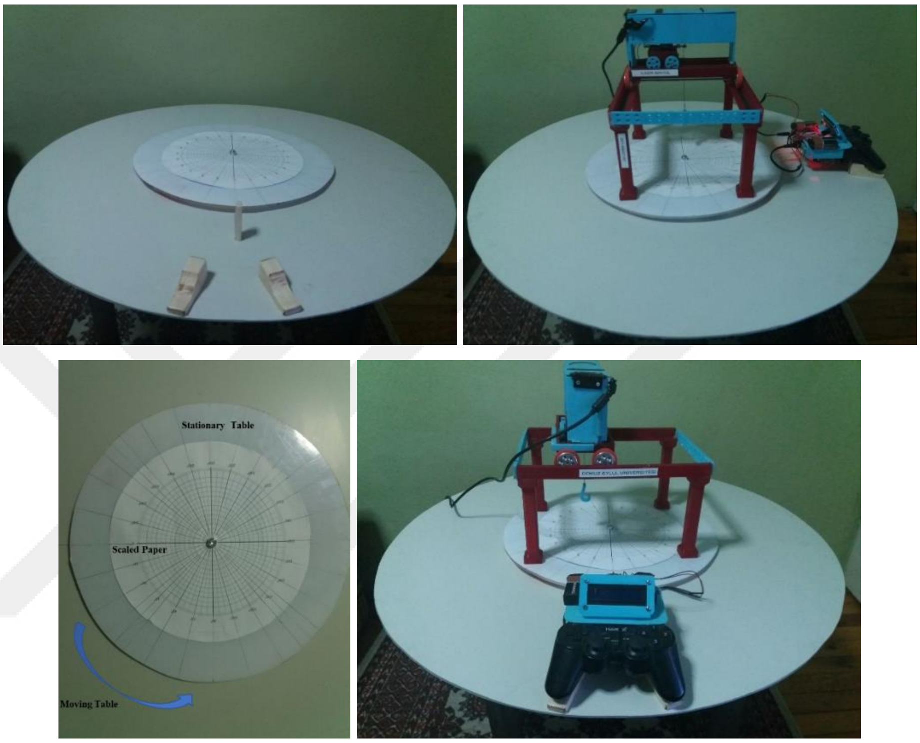


Figure 4.5 Experimental setup (Personal archive, 2018)

To test the performance of the designed radio remote control system the radio unit was moved around the crane from 0° to 360°. The performed experiments have shown that there is ± 2° deviation from the reference angle points from the operator position angels, which are given at Table 4.1. It is highly possible to have some undesirable movements if the crane starts operation at these critical angles. To have a safe crane operation, cranes can be prevented from starting initial movements at these critical angels or even a little bit wider than these critical angels. Moreover, to inhibit possible accidents, crane movement directions are not changed even the operator moves his/her position including these critical areas, provided that the operator goes on activating the joystick.

Table 4.1 Trolley and bridge movements related with operator position

Operator Position Angle	Joystick Movement	Trolley Movement	Bridge Movement
315 ° - 45 °	Right	+X Axis	-
	Back	-	-Z Axis
	Left	-X Axis	-
	Forward	-	+Z Axis
45 ° - 135 °	Right	-	-Z Axis
	Back	+X Axis	-
	Left	-	+Z Axis
	Forward	-X Axis	-
135 ° - 225 °	Right	-X Axis	-
	Back	-	+Z Axis
	Left	+X Axis	-
	Forward	-	-Z Axis
225 ° - 315 °	Right	-	+Z Axis
	Back	-X Axis	-
	Left	-	-Z Axis
	Forward	+X Axis	-

After experiment if we analyze the result of prototype crane, it can be easily seen that there is no any condition about y axis. Here, as seen from illustrations in Figure 4.1 y direction is related with hoist up and down directions. It is clear that hoist movements are independent of operator standing positions. All changings are performed at x and z axis movements of crane trolley and the bridge.

CHAPTER FIVE
CONCLUSIONS

For existing crane operation applications, there are unintentional situations which may cause the accidents and injuries. Generally, cranes are used to move very heavy materials in open or closed factory areas. In this application with standard hand-held control devices, the commands are fixed and they always activate the same motors at the same directions. In many cases, operators try to get the best position to follow up the moved material. When the operator changes his position, he has to be aware of changing commands in order to get correct movement of trolley and crane.

In this thesis a radio remote control prototype device has been designed to provide crane movements in respect to operator position. The prototype controller always checks the position of the operator in respect to the defined reference position and arrange the motor turning directions according to the current crane operator position. To calculate the relative movement an HMC5883L electronic compass was used. The communication between the transmitter and the receiver of the radio remote control unit was provided with NRF24L01 Wi-Fi module. The trolley and the crane bridge motors are accelerated with an L298N dual motor drive card.

As mentioned at experimental results there is about ± 2° deviation from the reference angle points. These critical points may cause unstable crane movements. To prevent unintentional movements of the crane, crane movements are not interrupted as long as the operator moves the joystick, even the operator changes his/her position. Moreover, the crane operator can be prevented from starting crane movements at critical points. For this, a warning symbol on the transmitter control unit can be illustrated to inform forbidden zones. These critical angles are the tolerances of HMC5883L electronic compass.

The designed prototype crane and the radio remote control unit have been tested and the obtained results have shown that the crane trolley and bridge moves in a way that the operator moves the joystick in respect to his/her standing position. It would be a good idea to place a switch at the transmitter of the radio transmitter unit to select or deselect this option. Either in this way operator can carry on with their standard habits or they enjoy the new safe radio control facility. This prototype system is certainly user friendly.

REFERENCES

3-Axis Digital Compass IC HMC5883L Retrieved February, 2013, from https://media.digikey.com/PDF/Data%20Sheets/Honeywell%20PDFs/HMC5883L.pdf

Colgate, J. E., Peshkin, M., & Klostermeyer, S. H. (2003). *Intelligent assist devices in industrial applications: a review. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* 2516-2521.

Han, J. Y. (2005). *Low-cost multi-touch sensing through frustrated total internal reflection. In Proceedings of the 18th annual ACM symposium on User interface software and technology* 115-118.

Interfaces and Control Systems for Intuitive Crane Control (2009). Retrieved December 21, 2017, from https://smartech.gatech.edu/bitstream/handle/1853/31782/peng_chen_chih_200912_mast.pdf

Kazerooni, H., & Guo, J. (1993). Human extenders. *Transactions-American Society of Mechanical Engineers Journal of Dynamic Systems Measurement and Control, 115*, 281-281.

Kazerooni, H., Fairbanks, D., Chen, A., & Shin, G. (2004). *The magic glove. In Robotics and Automation, 2004. Proceedings. ICRA'04.* 2004 IEEE International *Conference on* 1, 757-763.

Khalid, A., Huey, J., Singhose, W., Lawrence, J., & Frakes, D. (2006). *Human operator performance testing using an input-shaped bridge crane. Journal of dynamic systems, measurement, and control, 128*(4), 835-841.

Kivila, A., Porter, C., & Singhose, W. (2013). *Human operator studies of portable touchscreen crane control interfaces. In Industrial Technology (ICIT), 2013 IEEE International Conference on* 88-93.

Kivila, A., & Singhose, W. (2013*). Touchscreen crane control interfaces with oscillation suppression. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on* 5462-5467.

Lee, J. C., Hudson, S. E., Summet, J. W., & Dietz, P. H. (2005). *Moveable interactive projected displays using projector based tracking. In Proceedings of the 18th annual ACM symposium on User interface software and technology* 63-72.

Lee, J. C., Hudson, S. E., & Tse, E. (2008). *Foldable interactive displays. In Proceedings of the 21st annual ACM symposium on User interface software and technology* 287-290.

Merrill, D., Kalanithi, J., & Maes, P. (2007). Siftables: towards sensor network user interfaces. *In Proceedings of the 1st international conference on Tangible and embedded interaction* 75-78.

nRF24L01 Product Specification (2007). Retrieved December 21, 2017, from https://www.nordicsemi.com/eng/content/download/2730/34105/file/nRF24L01_Product_Specification_v2_0.pdf

Peng, C. C. (2009). *Interfaces and control systems for intuitive crane control*, (Phd thesis, Georgia Institute of Technology, USA).

Peshkin, M. A., Colgate, J. E., Wannasuphoprasit, W., Moore, C. A., Gillespie, R. B., & Akella, P. (2001). *Cobot architecture. IEEE Transactions on Robotics and Automation*, *17*(4), 377-390.

Raskar, R., Beardsley, P., van Baar, J., Wang, Y., Dietz, P., Lee, J., ... & Willwacher, T. (2004). *RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. In ACM Transactions on Graphics* (TOG) 23, 3, 406-415.

Sorensen, K. L. (2005). *A combined feedback and command shaping controller for improving positioning and reducing cable sway in cranes* (Doctoral dissertation, Georgia Institute of Technology).

Sorensen, K. L., Spiers, J. B., & Singhose, W. E. (2007). *Operational effects of crane interface devices. In Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on* 1073-1078.

Sorensen, K. L., Singhose, W., & Dickerson, S. (2007). *A controller enabling precise positioning and sway reduction in bridge and gantry cranes. Control Engineering Practice, 15*(7), 825-837.

Sorensen, K., Fisch, H., Dickerson, S., Singhose, W., & Glauser, U. (2008). *A multi-operational-mode anti-sway and positioning control for an industrial bridge crane. IFAC Proceedings Volumes, 41*(2), 881-888.

APPENDICES

APPENDICE-1-Receiver and Motor Control Arduino MEGA Codes

```
//Required library for NRF24L01
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
//Defining of variables
int rc[5];
// CE ve CSN fixed at  9 and 10 pins.
RF24 radio(9,10);
//Define NRF24L01 address.
const uint64_t baglanti = 0xE8E8F0F0E1LL;
// Define Bridge Motor Pins.
int motor1xpin = 22;
int motor1ypin = 23;
//Define Trolley Motor Pins
int motor2xpin = 24;
int motor2ypin = 25;
//Define Hoist Motor Pins
int motor3xpin = 4;
int motor3ypin = 5;
//Define motor speed.
int motor1hiz = 2;
int motor2hiz = 3;
int hizlanma = 35;
int vincuzunluk;
//Define Communication Control
int led=6;
int led2 = 7;
void setup(void) {
  //Begin NRF24L01 Wi-Fi Module
```

```
  radio.begin();
//Assigned this NRF24L01 Wi-Fi Module as a receiver.
  radio.openReadingPipe(1, baglanti);
  radio.startListening();
  //Define Output Pins.
  pinMode(led,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(motor1xpin, OUTPUT);
  pinMode(motor1ypin, OUTPUT);
  pinMode(motor2xpin, OUTPUT);
  pinMode(motor2ypin, OUTPUT);
  pinMode(motor3xpin, OUTPUT);
  pinMode(motor3ypin, OUTPUT);
  pinMode(motor1hiz, OUTPUT);
  pinMode(motor2hiz, OUTPUT);
}
void loop(void)
{
  if (radio.available())
  {
    bool done = false;
    while (!done)
    {
      delay(100);
      radio.read(rc, sizeof(rc));
      //Assign values which is read from NRF24L01 Wi-Fi Module ar “rc” series.
      int xPozisyonu = rc[0];
      int yPozisyonu = rc[1];
      int zPozisyonu = rc[2];
      int angle = rc[3];
      int baglanti = rc[4];
      if(baglanti == 100)
```

```
{
 digitalWrite(led,LOW);
 digitalWrite(led2,HIGH);
}
else
{
  digitalWrite(led2,LOW);
 digitalWrite(led,HIGH);
}
// Define motor turning direction and turn for North side.
if ((angle >= 0 && angle <= 45) || (angle >= 315 && angle <= 360) )      {
  if (xPozisyonu > 30 && yPozisyonu > 30)
  {
    gerisaga(hizlanma);
  }
  if (xPozisyonu < -30 && yPozisyonu > 30)
  {
    gerisola(hizlanma);
  }
  if (xPozisyonu > 30 && yPozisyonu < -30)
  {
    ilerisaga(hizlanma);
  }
  if (xPozisyonu < -30 && yPozisyonu < -30)
  {
    ilerisola(hizlanma);
  }
  if (yPozisyonu > 30 && (xPozisyonu < 30 && xPozisyonu > -30))
  {
    geriye(hizlanma);
  }
  if (yPozisyonu < -30 && (xPozisyonu < 30 && xPozisyonu > -30))
```

```
    {
     ileriye(hizlanma);
    }
    if (xPozisyonu < -30 && (yPozisyonu < 30 && yPozisyonu > -30))
    {
     sola(hizlanma);
    }
    if (xPozisyonu > 30 && (yPozisyonu < 30 && yPozisyonu > -30))
    {
     saga(hizlanma);
    }
    if ((xPozisyonu >= -30 && xPozisyonu <= 30 ) && (yPozisyonu >= -30 &&
yPozisyonu <= 30 ) )
    {
     dur();
    }    }
// Define motor turning direction and turn for East side.
   if ((angle >= 45 && angle <= 135))
   {
    if (xPozisyonu < -30 && yPozisyonu > 30)
    {
     gerisaga(hizlanma);
    }
    if (xPozisyonu < -30 && yPozisyonu < -30)
    {
     gerisola(hizlanma);
    }
    if (xPozisyonu > 30 && yPozisyonu > 30)
    {
     ilerisaga(hizlanma);
    }
    if (xPozisyonu > 30 && yPozisyonu < -30)
```

```
        {
         ilerisola(hizlanma);
        }
        if (yPozisyonu > 30 && (xPozisyonu < 30 && xPozisyonu > -30))
        {
         saga(hizlanma);
        }
        if (yPozisyonu < -30 && (xPozisyonu < 30 && xPozisyonu > -30))
        {
         sola(hizlanma);
        }
        if (xPozisyonu < -30 && (yPozisyonu < 30 && yPozisyonu > -30))
        {
         geriye(hizlanma);
        }
        if (xPozisyonu > 30 && (yPozisyonu < 30 && yPozisyonu > -30))
        {
         ileriye(hizlanma);
        }
        if ((xPozisyonu >= -30 && xPozisyonu <= 30) && ( yPozisyonu >= -30 &&
yPozisyonu <= 30 ) )
        {
         dur();      }     }
// Define motor turning direction and turn for South side.
      if ((angle >= 135 && angle <= 225))
      {
        if (xPozisyonu < -30 && yPozisyonu < -30)
        {
         gerisaga(hizlanma);
        }
        if (xPozisyonu > 30 && yPozisyonu < -30)
        {
```

```
  gerisola(hizlanma);
}
if (xPozisyonu < -30 && yPozisyonu > 30)
{
  ilerisaga(hizlanma);
}
if (xPozisyonu > 30 && yPozisyonu > 30)
{
  ilerisola(hizlanma);
}
if (yPozisyonu > 30 && (xPozisyonu < 30 && xPozisyonu > -30) )
{
  ileriye(hizlanma);
}
if (yPozisyonu < -30 && (xPozisyonu < 30 && xPozisyonu > -30))
{
  geriye(hizlanma);
}
if (xPozisyonu < -30 && (yPozisyonu < 30 && yPozisyonu > -30))
{
  saga(hizlanma);
}
if (xPozisyonu > 30 && (yPozisyonu < 30 && yPozisyonu > -30))
{
  sola(hizlanma);
}
if ((xPozisyonu >= -30 && xPozisyonu <= 30) && (yPozisyonu >= -30 &&
yPozisyonu <= 30) )
{
  dur();
}      }
```

```
// Define motor turning direction and turn for West side.
    if ((angle >= 225 && angle <= 315))
    {
     if (xPozisyonu > 30 && yPozisyonu < -30)
     {
      gerisaga(hizlanma);
     }
     if (xPozisyonu < - 30 && yPozisyonu < - 30)
     {
      ilerisaga(hizlanma);
     }
     if (xPozisyonu > 30 && yPozisyonu > 30)
     {
      gerisola(hizlanma);
     }
     if (xPozisyonu < -30 && yPozisyonu > 30)
     {
      ilerisola(hizlanma);
     }
     if (yPozisyonu > 30 && (xPozisyonu < 30 && xPozisyonu > -30))
     {
      sola(hizlanma);
     }
     if (yPozisyonu < -30 && (xPozisyonu < 30 && xPozisyonu > -30))
     {
      saga(hizlanma);
     }
     if (xPozisyonu < -30 && (yPozisyonu < 30 && yPozisyonu > -30))
     {
     ileriye(hizlanma);
     }
     if (xPozisyonu > 30 && (yPozisyonu < 30 && yPozisyonu > -30))
```

```
        {
          geriye(hizlanma);
        }
        if ((xPozisyonu >= -30 && xPozisyonu <= 30) && ( yPozisyonu >= -30 &&
yPozisyonu <= 30) )
        {
          dur();        }      }
// Define hoist system motor turning direction.
      if (zPozisyonu == 10 ) // eğer zPozisyonu 10 ise yukarı doğru çevir
      {
        yukari();
        delay(50);
      }
      else if (zPozisyonu == -10 ) //eğer Zposizyonu -10 ise aşağı doğru çevir
      {
        asagi();
        delay(50);
      }
      else if (zPozisyonu == 0) //Eğer 0 ise durdur
      {
        digitalWrite(motor3xpin, LOW);
        digitalWrite(motor3ypin, LOW);
      }
    }
  }
  else
  {
    digitalWrite(led,HIGH);
    digitalWrite(led2,LOW);
  }
}
```

APPENDICE-2-Transmitter and Compass Arduino UNO Codes

```
//Required library for NRF24L01
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
// Required library for  HMC5883
#include <Wire.h>
#include <math.h>
#include <Adafruit_HMC5883_U.h>
#include <Adafruit_Sensor.h>
// Define Compass.
Adafruit_HMC5883_Unified compass = Adafruit_HMC5883_Unified(12345);
//CE pin set as 9 and CSN pin set as 10 at NRF24L01
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;  //Radio Frequency Address Assingnment
//Define 5 variable serial.
int rc[5];
//Crane hoist system movement arrangements define at pin2 and pin 3.
int yukari=2;
int asagi = 3;
//Define analog inputs which is used joystick for crane trolley movement.
int xPin = A0;
int yPin = A1;
//Analog values change between 0-1023 but we will change the vsalues 0-255 with map command.
 xPozisyonu = 0;
int yPozisyonu = 0;
int zPozisyonu = 0; // For hoist system up and down movement variable.
int gonderilecekaci = 0; // This line written because of not any directional change while operator movement.
int angle;
 // The angle came from the HMC5883.
```

```
 void setup(void){
//2. ve 3. Pin set as  input.
pinMode(2,INPUT);
pinMode(3,INPUT);
//Begin Compass
compass.begin();
Wire.begin();
Serial.begin(9600);
//Begin NRF24L01
radio.begin();
radio.openWritingPipe(pipe);}
void loop(void){
//The angel came from HMC5883 changed with this code as a radial angle.
sensors_event_t event;
compass.getEvent(&event);
float heading = atan2(event.magnetic.y, event.magnetic.x);
float declinationAngle = 0.22;
if(heading < 0) heading += 2*PI; if(heading > 2*PI)heading -= 2*PI;
float headingDegrees = heading * 180/M_PI;
angle=(int)headingDegrees;
//Define variables for Anolog Inputs.
xPozisyonu = analogRead(xPin);
yPozisyonu = analogRead(yPin);
 if (xPozisyonu<=512)
    {
      rc[0]=map(xPozisyonu,0,512,125,0);
    }
  if (xPozisyonu>512)
    {
      rc[0]=map(xPozisyonu,512,1023,0,-125);
    }
  if (yPozisyonu<=512)
```

```
  {
    rc[1]=map(yPozisyonu,0,512,125,0);
  }
if (yPozisyonu>512)
  {
    rc[1]=map(yPozisyonu,512,1023,0,-125);

//Define Up and Down buttons.
int yukari_deger = digitalRead(yukari);
int asagi_deger = digitalRead(asagi);
if(yukari_deger == HIGH)
  {
    zPozisyonu=10;
  }
if(asagi_deger == HIGH)
  {
    zPozisyonu=-10;
  }
if(asagi_deger==LOW && yukari_deger == LOW)
  {
    zPozisyonu=0;
  }
rc[2] = zPozisyonu;
if( rc[0] > 30 )
  {
    rc[3] = gonderilecekaci;
  }
else if( rc[0] < -30 )
  {
    rc[3] = gonderilecekaci;
  }
else if( rc[1] > 20 )
```

```
    {
      rc[3] = gonderilecekaci;
    }
  else if( rc[1] < -20 )
    {
      rc[3] = gonderilecekaci;
    }
   else if (rc[1] > -20 || rc[1] < 20 && rc[0] > -20 || rc[0] <20 )
    {
      gonderilecekaci = angle;
      rc[3] =gonderilecekaci;
    }
    rc[4] =100;
radio.write(rc,sizeof(rc));  }
```

APPENDICE- 3-LCD Panel Arduino Uno Codes

LCD Panel Arduino Uno Codes

```
//Required library for LCD Panel
#include <AvrI2c_Greiman.h>
#include <LiquidCrystal_I2C_AvrI2C.h>
//Define LCD
LiquidCrystal_I2C_AvrI2C lcd(0x3F,16,2);
//Define joystick pins and buttons, create variables.
int joystick[2];
int xPin = A0;
int yPin = A1;
int xPozisyonu = 0;
int yPozisyonu = 0;
int yukari =2;
int asagi = 3;
int lcdackapa = 4;
int lcdbaslama = 5;
void setup() {
  Serial.begin(9600);
  //Start LCD and define input buttons.
  lcd.begin();
  pinMode(lcdackapa,INPUT);
  pinMode(yukari,INPUT);
  pinMode(asagi,INPUT);
  lcdbasla();
}
void loop() {
//Working LCD panel permenant.
  anaekran();
  delay(300);
  //Read anolog values and assigned to variables.
  xPozisyonu = analogRead(xPin);
```

```
yPozisyonu = analogRead(yPin);
//Joystick positions limited between -125 and 125.
joystick[0]=map(xPozisyonu,0,1023,-125,125);
joystick[1]=map(yPozisyonu,0,512,125,0);
if(joystick[0] < -30)
  {
   lcd.setCursor(3,0);
   lcd.print(" ");
   delay(200);
   lcd.setCursor(6,0);
   lcd.print("ileri");
  }
if(joystick[0] > 30)
 {
   lcd.setCursor(3,1);
   lcd.print(" ");
   delay(200);
   lcd.setCursor(6,0);
   lcd.print("geri");
  }
if(joystick[1] > 30)
 {
   lcd.setCursor(2,0);
   lcd.print(" ");
   lcd.setCursor(2,1);
   lcd.print(" ");
   delay(200);
   lcd.setCursor(7,1);
   lcd.print("sola");
  }
if(joystick[1] < -30)
 {
```

```
    lcd.setCursor(4,0);
    lcd.print(" ");
    lcd.setCursor(4,1);
    lcd.print(" ");
    delay(200);
    lcd.setCursor(7,1);
    lcd.print("saga");
   }
if(digitalRead(yukari)==HIGH)
{
  lcd.setCursor(13,0);
  lcd.print(" ");
  delay(200);
  if(joystick[0] < 30 && joystick[0] >-30)
  {
    lcd.setCursor(6,0);
    lcd.print("yukari");
  }
}
else if(digitalRead(asagi)==HIGH)
{
  lcd.setCursor(13,1);
  lcd.print(" ");
  delay(200);
  if(joystick[1] < 30 && joystick[1] >-30)
  {
    lcd.setCursor(6,1);
    lcd.print("asagi");
  }
}
//If there is no command, all letters disappear with this command.
  if( joystick[1] <30 && joystick[1] >-30 && digitalRead(asagi)==LOW)
```

```
 {
  lcd.setCursor(6,1);
  lcd.print("      ");
 }
 if( joystick[0] <30 && joystick[0] >-30 && digitalRead(yukari)==LOW)
 {
  lcd.setCursor(6,0);
  lcd.print("      ");
  }
}
//Creating Cursors
void anaekran()
{
 karakterolustur();
 // Forward Cursor
 lcd.setCursor(2,0);
 lcd.write((uint8_t)5);
 //Back Cursor
 lcd.setCursor(2,1);
 lcd.write((uint8_t)6);
 //Left Cursor
 lcd.setCursor(3,0);
 lcd.write((uint8_t)0);
 lcd.setCursor(3,1);
 lcd.write((uint8_t)1);
 //Right Cursor
 lcd.setCursor(4,0);
 lcd.write((uint8_t)4);
 lcd.setCursor(4,1);
 lcd.write((uint8_t)3);
 //Up Cursor
 lcd.setCursor(13,0);
```

```
  lcd.write((uint8_t)0);
  //Down Cursor
  lcd.setCursor(13,1);
  lcd.write((uint8_t)1);
  delay(10);
}
void lcdbasla()
{
  lcd.clear();
//Define direction cursors.
  karakterolustur();
//Starting Our characters at first line second column.
  lcd.setCursor(2,0);
  lcd.print("HOSGELDINIZ");
  delay(2000);//
  lcd.clear();//
  lcd.setCursor(3,0);
  lcd.write((uint8_t)7);
  lcd.setCursor(4,0);
  lcd.print("lker");
  lcd.setCursor(7,1);
  lcd.print("B");
  lcd.setCursor(8,1);
  lcd.write((uint8_t)7);
  lcd.setCursor(9,1);
  lcd.print("RYOL");
  delay(1000);
  lcd.clear();
}
byte yukariok[8] = {
  0b00100,
  0b01010,
```

```
  0b10101,
  0b01010,
  0b10001,
  0b00000,
  0b00000,
  0b00000
};
byte asagiok[8] = {
  0b00000,
  0b00000,
  0b00000,
  0b10001,
  0b01010,
  0b10101,
  0b01010,
  0b00100
};
byte sagust[8] = {
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b10100,
  0b01010,
  0b00101
};
byte sagalt[8] = {
  0b00101,
  0b01010,
  0b10100,
  0b00000,
```

```
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
byte solust[8] = {
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00101,
  0b01010,
  0b10100
};
byte solalt[8] = {
  0b10100,
  0b01010,
  0b00101,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
byte iharfi[8] = {
0b00100,
  0b01110,
  0b00100,
  0b00100,
  0b00100,
  0b00100,
```

```
  0b01110,
  0b00000
};
byte indir[8] = {
  0b00000,
  0b00000,
  0b01110,
  0b01110,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};
byte cikar[8] = {
  0b00000,
  0b00100,
  0b01110,
  0b11111,
  0b01110,
  0b01110,
  0b00000,
  0b00000
};
void karakterolustur()
{
  lcd.createChar(0, yukariok);
  lcd.createChar(1, asagiok);
  lcd.createChar(3,sagalt);
  lcd.createChar(4,sagust);
  lcd.createChar(6,solalt);
  lcd.createChar(5,solust);
  lcd.createChar(7,iharfi);   }
```

APPENDICE-4-Symbols and Abbreviations

PID: Proportional Integral Derivative
AC: Alternative Current
LHD: Load Haul Dump
HCI: Human Computer Interface
UI: User Interface
GUI: Graphical User Interface
IR: Infra-Red
LED: Light Emitting Diode
VR: Virtual Reality
RF-ID: Radio Frequency Identification
RTLS: Real Time Location System
DC: Direct Current
RC: Radio Control
PWM: Pulse Width Modulation
ICSP: In Circuit Serial Programming
GND: Ground
V: Voltage (volt)
A: Ampere (ampere)
mA: Milliampere (ampere)
TTL: Transistor-Transistor-Logic
MISO: Master In Slave Out
MOSI: Master Out Slave In
SPI: Serial Peripheral Interface
SCK: Serial Clock
TWI: Two Wire Interface
SDA: Data Line
SCL: Clock Line
PCB: Printed Circuit Board
mm: Millimeter (meter)
cm: Centimeter (meter)
RPM: Revolutions Per Minute
GHz: Giga-Hertz
FIFO: First In Firs Out
MCU: Micro Controller Unit
GFSK: Gaussian Frequency Shift Keying
***Kbps*:** Kilobit Per Second
***Mbps*:** Megabit Per Second
ADC: Analog to Digital Converter
LCC: Leadless Chip Carrier