National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

EE-423: EMBEDDED SYSTEM DESIGN

SEMESTER PROJECT

PROJECT TITLE: Hand Gesture Based TV Remote

CLASS: BEE10

GROUP: ESD Gp-2

INSTRUCTOR: Dr. Usman Zabit

SUBMITTED BY:

→Muhammad Asad Imran Rafique (reg#247556)

→Ameer Moavia (reg#248021)

→Ali Abdul Rehman (reg#247282)

DEDICATION:

We hereby dedicate this dissertation towards the cause of putting our engineering studies to practical use. The project is a hand gesture recognition system that replaces the TV remote control. Gesture recognition is an alternative user-interface for providing real-time data to a device.

We further dedicate this project to our families, our teachers and our fellow students who helped us along the way. This would not be possible without the support we received.

The students aim to utilize the skills learned in embedded system design and adapt them to create a practical every-day use device.

Finally, we would like to thank the Almighty, to whom belongs all knowledge and wisdom.

ACKNOWLEDGEMENTS:

We would like to acknowledge the support we received from our instructor, Dr. Usman Zabit, to pursue this project. Through his teachings, we received a vast influx of ideas of how to implement our design.

We would also like to acknowledge the aid we received from our classmates, who helped us whenever we asked and made this project possible.

We would finally like to thank NUST and SEECS for providing us with an environment suitable for working on the project as well as providing resources to help bring the project to life.

ABSTRACT:

Hand gestures have been removing dependencies on remote devices for the past few years. The aim of this project is to accelerate this removal. Hand gestures are a simple and natural way of giving instructions so the logical step would be to instruct technology using them.

TVs and other household gadgets have been using Infrared remotes for a very long time. There may be some devices with gesture-recognition ability however, these devices are a burden on the common man's pocket.

The scope of this project will be to implement a hand-gesture based TV remote with user friendly controls, while also being cost effective for the user. The project will use a pair of Ultrasonic sensors on a master controller which will receive a hand gesture in real-time and decide the command. The master will wirelessly send the information to the slave controller which will convert the command to the relevant IR signal, to be transmitted to the TV.

In future work, we can enhance the project to incorporate several different devices and a huge range of gestures at same time however that is beyond the current scope of our project.

Key words-

Hand Gesture, real-time, sensors, controllers, user-friendly, wireless

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1: Objectives

The project was aimed to design a real-time hand-gesture based tv remote control. The scope of this project covered the knowledge of real-time operating systems and connected sensor networks that we have learnt through the course of embedded system design.

The major objective was to remove dependency on handheld devices for convenience's sake. It is easier to communicate via gestures than pressing buttons on a remote. Adding to this is the ongoing pandemic which has given rise to contactless systems for hygiene's sake. The project, if implemented, can aid in changing tv settings in a public area without remote controls. Finally, we aim to ensure proper functionality through effective multitasking.

The project features include a user-friendly interface using simple hand waving gestures. A simple interface however does not mean simple programming. The system was brought to life with arduous efforts. Majority of it on brainstorming solutions that would be cost effective and practical. The product uses 2 ultrasonic sensors to receive hand-gestures in real time at the mother node and transmits the relayed command to the daughter node to send an IR signal to the tv. The two nodes are placed in such a way that the mother node is close to the user and the daughter node is close to the tv receiver. This requires wireless transmission between both nodes.

The project is aimed to show how one can use embedded systems to solve daily-life problems. The complexity of the system can be increased however it is beyond the scope of the current project. This project simply shows one possible way.

1.2: Block diagram of complete system

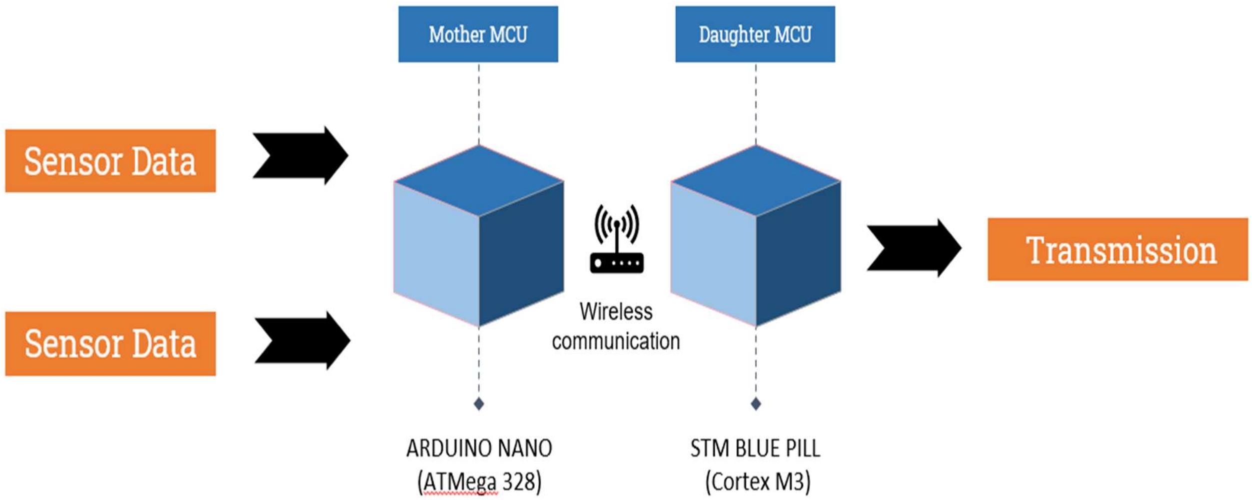

Figure 1: Block Diagram

CHAPTER 2: DESIGN METHODOLOGY

2.1: Problem Statement

The situation at hand was to design a real-time system of connected sensors and actuators that receives user input via hand-gestures at one node and transmits an IR signal at another node. The two nodes must communicate wirelessly so that they may be placed at optimal locations. Extra design requirements include beautification and cost control.

2.2: Analysis of Ideas

At the start, the main idea was to implement a computer-vision based system using open-source hand gesture libraries and a camera module connected to the mother node. However, this was rejected once it seemed to be too much work with not much gain. The controllers we had opted to utilize were not powerful enough. It was nearly impossible to program gesture-recognition via vision correctly and the setup became complicated. Therefore, it was agreed to use two ultrasonic sensors at the mother node for gesture reception. In theory, this project can be extended to incorporate any number of devices and gestures. However, the downside is that the number of gestures each device requires is exponentially increased and it will become difficult to memorize gestures for the user. For this project, we only used 1 device and incorporated power, volume, and channel features with 5 gestures altogether. The main intent of the project is to prove a cost-effective alternative to already existing gesture-recognition based devices.

2.3: Develop Solutions

The Hand gesture-based TV remote control system should use simple hand gestures that are easy to detect with ultrasonic sensors. Therefore, we decided to use forward and backward hand motion for either hand to send commands. The maximum range for the distance between sensor and hands was kept at 60 cm. There was an idea to implement sleep mode for power saving, but it was scrapped due to time constraints and because the MCUs we used did not require no power saving. We chose Arduino-Nano to be the mother node and STM32 Blue-pill for the daughter node. Both devices consume very low power and have a simple programming process.

The mother node had to continuously read the real-time data coming from the sensors and process it to determine what command is being relayed by the user. Then, via RF transceiver module, it would send the command to the daughter node. The daughter node would receive the command via RF transceiver module and send the equivalent IR signal to the TV.

A major process was to acquire the IR signals to control a TV. This task was fulfilled by a stand-alone node with only an IR receiver. The raw data was collected for each button that we would be using for our project and used later by the daughter node. This node is however not part of the complete system and is only useful for configuration of a new remote.

2.4: Mother Node Design

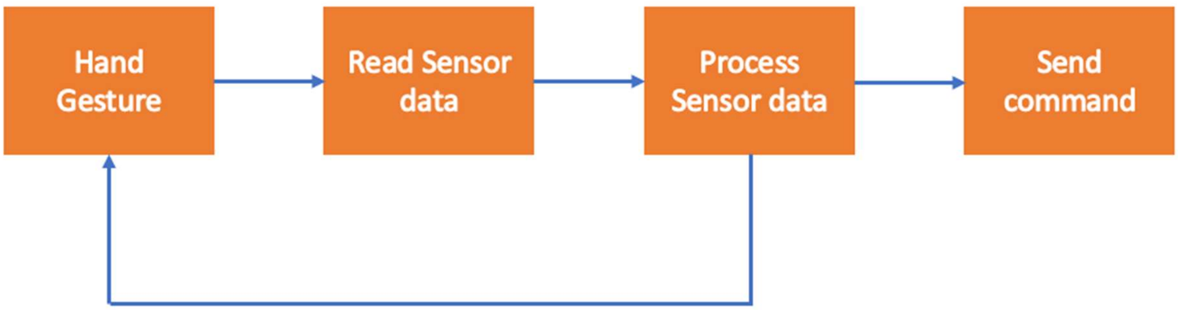

Figure 2: Mother Node Design

2.5: Daughter Node Design

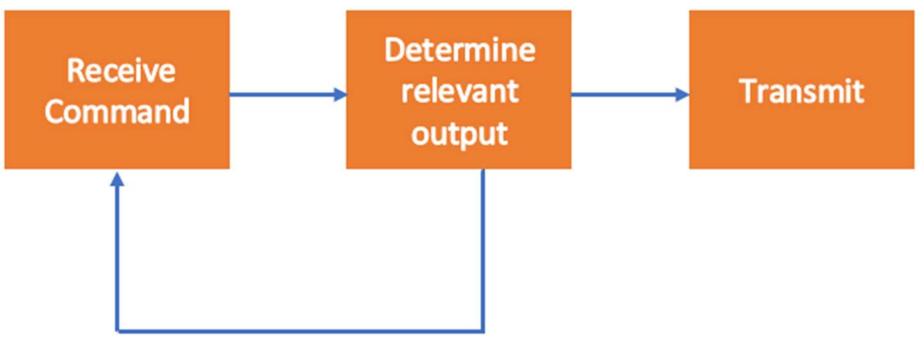

Figure 3: Daughter Node Design

2.6: IR signal Acquisition Node Design

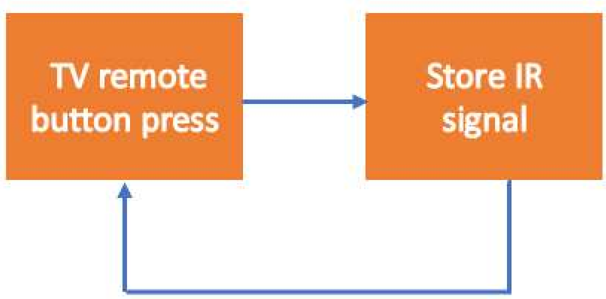

Figure 4: IR signal Acquisition Node Design

2.7: Complete System Design

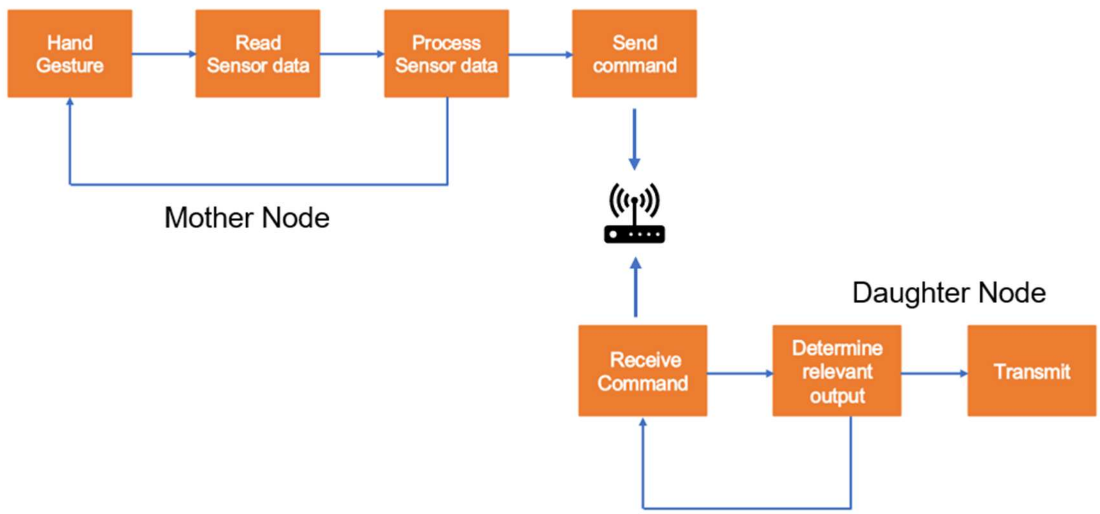

Figure 5: Complete System Design

CHAPTER 3: MCU SPECIFICATIONS

3.1: Transmitter (Mother) MCU

The Transmitter (mother) MCU is an Arduino Nano (ATMEGA328P) which is linked with 2 ultrasonic sensors (left and right), as well as an RF transceiver module. The sensors detect hand movement and relay the corresponding command to the daughter MCU via RF link. For example, if the user is to place both hands at a distance between 40-60 cm from the sensors, then the command to power on or off the TV will be sent. There is a no-command option which is sent when no gesture is detected. The transmitter MCU model is shown below.

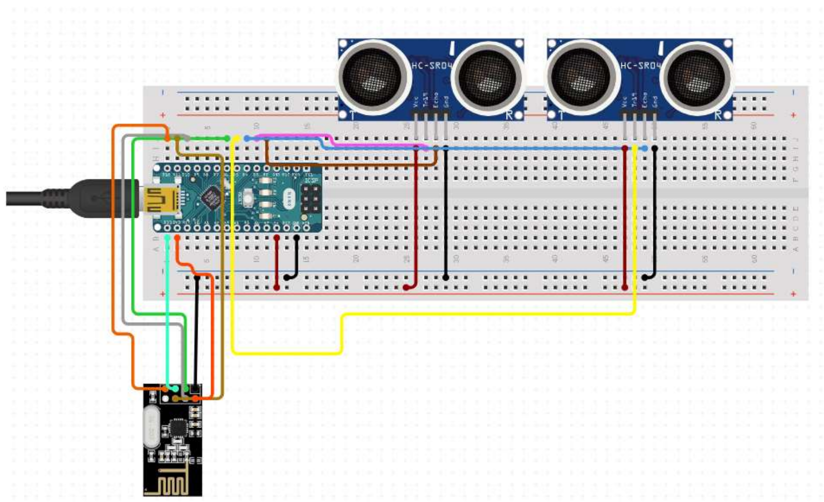

Figure 6: Transmitter (Mother) Node Schematic

3.2: Receiver (Daughter) MCU

The Receiver (daughter) MCU is a STM32 blue pill which is attached to an RF transceiver module and an IR LED (a transistor and resistor are needed for voltage control). This node is constantly receiving commands from the mother MCU. The corresponding IR signal is then relayed to the IR LED to control the TV. The receiver MCU model is shown below.

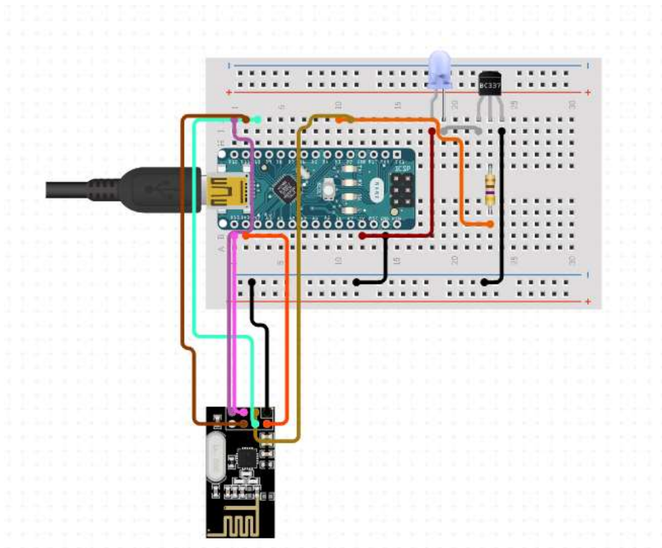

Figure 7: Receiver (Daughter) Node Schematic

3.3: IR signal Acquisition MCU

The IR signal acquisition MCU is an Arduino Nano which is attached only to an IR receiver module. This MCU is loaded with an example code from an open-source Arduino library for IR remote. The raw data for relevant button presses is stored in a text file and used in the daughter MCU. The acquisition MCU model is shown below.

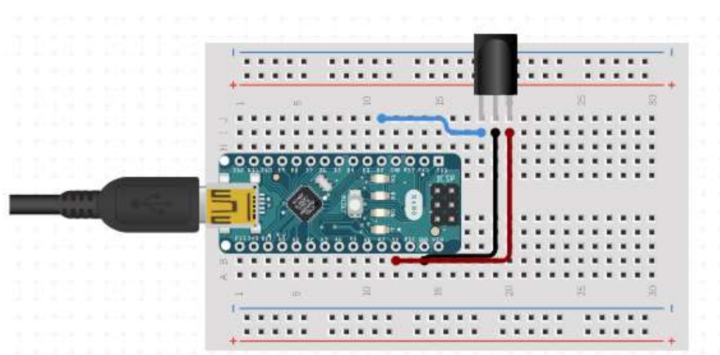

Figure 8: IR signal acquisition Node Schematic

3.4: List of components used

S No.	Component Name	Number used
1.	Arduino Nano	2
2.	STM 32 Blue-pill	1
3.	nRF24L01 RF transceiver	2
4.	HC-SR04 Ultrasonic Sensor	2
5.	IR Transmitter	1
6.	IR receiver	1

Figure 9: List of components

CHAPTER 4: USE OF FEATURES OF FREERTOS

The project uses FreeRTOS in the mother MCU because there was a need to acquire real-time sensor data and transmit processed data to another node. The following features of FreeRTOS were implemented in the project.

4.1: Semaphore (Mutexes)

The Transmitter (mother) MCU makes use of mutexes for the serial and radio port. The serial port mutex helps with debugging only whereas the radio port ensures that the port is free to use by the sending task.

```
SemaphoreHandle_t xSerialSemaphore; // Declare a mutex Semaphore Handle which we will use to manage the Serial Port.
SemaphoreHandle_t xRadioSemaphore; // Declare a mutex Semaphore Handle which we will use to manage the Radio Port.
```

Figure 10: Declaring Serial Port and Radio Port Semaphore (Mutex) handles

```
if ( xSerialSemaphore == NULL )  // Check to confirm that the Serial Semaphore has not already been created.
{
  xSerialSemaphore = xSemaphoreCreateMutex();  // Create a mutex semaphore we will use to manage the Serial Port
  if ( ( xSerialSemaphore ) != NULL )
    xSemaphoreGive( ( xSerialSemaphore ) );  // Make the Serial Port available for use, by "Giving" the Semaphore.
}
```

Figure 11: Creating and freeing Serial Port Mutex

```
if ( xRadioSemaphore == NULL )  // Check to confirm that the Radio Semaphore has not already been created.
{
  xRadioSemaphore = xSemaphoreCreateMutex();  // Create a mutex semaphore we will use to manage the Radio Port
  if ( ( xRadioSemaphore ) != NULL )
    xSemaphoreGive( ( xRadioSemaphore ) );  // Make the Radio Port available for use, by "Giving" the Semaphore.
}
```

Figure 12: Creating and freeing Radio Port Mutex

4.2: Queues

The Transmitter (mother) MCU uses both integer queues and struct queues which are used for inter-task communication.

```
QueueHandle_t distanceQueue; // declare a Handle for Queue to store distance of both sensors (struct queue)
QueueHandle_t commandQueue; // declare a Handle for Queue to store command to be sent through radio (int queue)
```

Figure 13: Declaring distance and command Queue Handles

```
distanceQueue = xQueueCreate(2, sizeof(struct distance));
commandQueue = xQueueCreate(10, sizeof(int));
```

Figure 14: Creating distance (struct) queue and command (integer) queue

4.3: Task Pre-emption Algorithms

The Transmitter (mother) MCU uses task pre-emption algorithms to effectively perform its functions. The Send data task has the highest priority whereas the read sensor data task has the lowest priority. This is because sending data to the daughter MCU is the most important task.

```
if (distanceQueue != NULL) {
  xTaskCreate(
    TaskReadSensorData
    ,  "Get Sensor Data"
    ,  128
    ,  NULL
    ,  1  // Lowest priority task
    ,  NULL );

  xTaskCreate(
    TaskProcessData
    ,  "process sensor data"
    ,  128
    ,  NULL
    ,  2  // Medium priority task
    ,  NULL );

  xTaskCreate(
    TaskSendData
    ,  "send sensor data"
    ,  128
    ,  NULL
    ,  3  // Highest priority task
    ,  NULL );
}
```

Figure 15: Defining priorities to tasks

CHAPTER 5: CONCLUSIONS

5.1: Hardware issues/results/observations

The initial problem we faced was to find a way to program STM32 blue-pill cortex M3 device on Arduino IDE rather than bare-metal coding. It was solved through researching and using multiple STM32-arduino cores after which we settled on the perfect one.

We faced a few problems while dealing with hardware. Our jumper wires were faulty at times and our sensors gave garbage values. The voltage supply was supposed to be 5 V, and this resulted in an Arduino Nano burning up. After this, we used an external power supply for the sensors with voltage regulation.

The most challenging part of this project was to create the processing sensor data task to accurately detect commands sent by the user. This issue was resolved with the use of queues to get previous and current sensor data and to keep overwriting them periodically. The comparison of previous and current data from left and right sensors can give precise reading of the command that the user wants to send.

The project uses modern embedded systems technology to improve quality of life. It is a multi-tasking system that can detect a wide variety of gestures and transmit commands using FreeRTOS.

5.2: Future Recommendations

The project can be upgradeable for any system that uses IR signals for example ac remotes. Furthermore, it can be configured for controlling multiple devices simultaneously using different gestures, essentially creating a contactless universal remote.

REFERENCES

[1] A. Raj, "CircuitDigest," 24 10 2017. [Online]. Available: https://circuitdigest.com/microcontroller-projects/control-your-computer-with-hand-gestures.

[2] Y. Lei, W. Hongpeng, T. Dianxiong and W. Jue, "A real-time hand gesture recognition algorithm for an embedded system," IEEE, Tianjin, China, 2014.