

**Program: ESE 4009**

**INSTRUCTOR:** Prof**.** Mike Aleshams

# Group 1

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Signature\* |
| Navjot Saini | C0751275 | Navjot Saini |
| Ramneet kaur | C0752942 | Ramneet Kaur |

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties.*

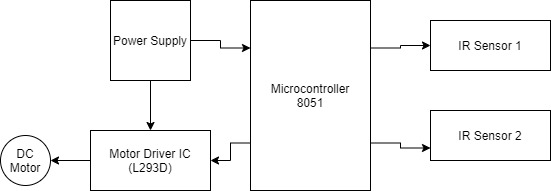
**Project Proposal**

**Project Title: IoT based Automatic Railway Gate Controller**

**Description of the latest similar system:**

# Automatic Railway Gate Controller

Automatic Railway Gate Control System is a simple but very useful project, which help is automatically opening and closing the railway gate upon detecting arrival or departure of the train.The principle of operation behind the working of this project lies in the functioning of IR Sensor. A Reflective type IR Sensor is used in this project.In Reflective Type IR Sensor, the IR transmitter and receiver are placed side by side. When there is no obstacle in front of the sensor, the IR rays transmitted by the IR Transmitter will travel undetected as there are no rays falling on the IR Receiver.If there is an obstacle in front of the IR Transmitter and Receiver pair, the IR Rays gets reflected off from the surface of the obstacle and are incident on the IR Receiver.This setup can be configured to detect an object like a Train and in turn can be used to switch ON or OFF the loads like motors with the help of microcontroller.



***Figure 1: Block Diagram of Automatic Railway Gate Controller***

**Working Principle of the Project:**

Practically, the two IR sensors are placed at left and right side of the railway gate. The distance between the two IR sensors is dependent on the length of the train. In general, we must consider the longest train in that route.If the sensor 1 detects the arrival of the train, microcontroller starts the motor with the help of motor driver in order to close the gate.The gate remains closed as the train passes the crossing.When the whole train crosses the gate and reaches second sensor, it detects the train and the microcontroller will open the gate.

**Limitations of the latest similar system:**

1. The microcontroller 8051 is 8-bit controller which is slow in processing as compared to advanced processor.

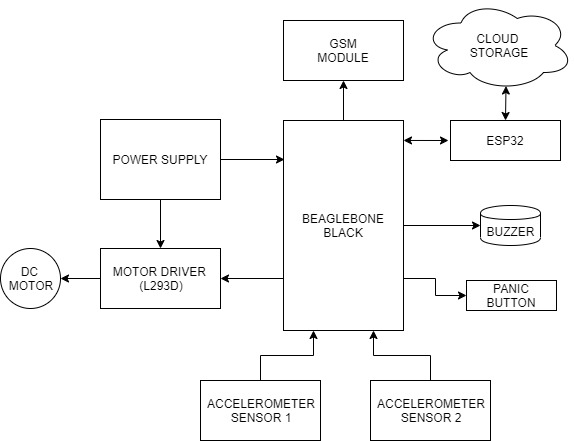
2. System use IR sensors which have certain limitations with respect to range and environment.

3. No implementation is done in case of emergency.

4. No measurement of the speed of the train at which the train is arriving.

5. Lack of IoT Technology.

**Solution 1:**

****

***Figure 2: Block Diagram for solution 1 of IoT based Automatic Railway Gate Controller***

**Features of Solution 1:**

* Use of BeagleBone Black as master device which is 32-bit microprocessor with high processing speed.
* Use of IoT through Wi-Fi for cloud storage, such as,ThingSpeak.
* **Use of various peripherals such as touch screens, cameras, microphones and speakers, GPIOs, timers, GPS modules, Bluetooth, WiFi, and ADC/DACs?**
* Use of ESP32 module which has Wi-Fi and Bluetooth capabilities usingIEEE802.11 protocol.
* Use of GSM Module to send alert message to the authorities in case of emergency situation.
* Use of DC motor interfaced with DC motor driver circuit L293D for closing and opening of gates.
* Use of the accelerometer sensor to monitor the vibrations of the railway track.
* Use of Panic Button in an emergency situation to alert the authorities in case of any mishappening around tracks through GSM module.
* Use of Buzzer to alert the users about upcoming trains and closure of gates.
* **Use of I2C, SPI, RS232/RS-485, IrDA infrared, JTAG, USB, Bluetooth, IEEE 802.11 WiFi, IEEE 802.3 Ethernet, CAN and GPS protocols and systems?**
* The Wi-Fi protocol used is IEEE802.11.
* The protocol used for communicating with cloud is MQTT.
* L293D motor driver IC is used to interface DC motor with Beaglebone Black and it is connected to GPIO pins of BBB.
* Wi-Fi module ESP32 supports SPI, I2C, UART, ADC and DAC protocols with the help of GPIO pins.
* GPIO pins of BeagleBone Black is used to interface Buzzer.
* Panic Button is directly interfaced with GPIO pins of BBB.
* I2C interfacing between Accelerometer sensor and BeagleBone Black.
* Interfacing of GSM module with BBB can be done using serial communication protocol.
* **Use of preemptive versus cooperative scheduler operation; tick rate and time slicing; critical code; fixed, dynamic and hybrid task priority allocation; application-specific considerations; power management tactics; semaphores, mutexes and queues; debugging strategies; performance estimation?**
* The system uses Real time concepts such as the pre-emptive scheduler operation wherein each task related to sensors and other peripherals will be assigned priorities for their functioning.
* The Panic button feature will act as an interrupt thus will have critical code for that section.
* Since the system uses pre-emptive scheduler, thus the tasks will be time-sliced else it will non pre-emptive in nature.
* Also, due to some conflicts of the resources may occur, thus we may employ the semaphores/ binary semaphores (i.e. mutexes) so that each task that executes will have to obtain the key before the execution.

**Software and Hardware Requirements for Solution 1:**

**Software Requirements:**

1. Debian Latest Image for the Beaglebone Black
2. Easy EDA/ Fritzing/ Proteus ISIS
3. ThingSpeak for Cloud Storage
4. C Programming/ Python
5. Eclipse IDE
6. GCC Compiler or any cross compiler

**Software Tools:**

1. **Text editor:** To write source code in programming languages C and C++ and save this code as a text file, Nano, Vim text editors will be used in the project.
2. **Integrated Development Environment (IDE):** Eclipse IDE for programming in C language as  software that provides a set of necessary tools in one package.Arduino IDE to write program for Arduino Uno.
3. **Compiler:** A compiler is a tool for transforming the code into a low-level machine language code. GCC Compiler or any cross compiler such as Linaro can be used in this project.Other compiler that we can use in this project is ARM GCC Cross Compiler Toolchain.
4. **Simulation Software:**Simulation is the process of designing a model of a real system.Easy EDA/ Fritzing/Proteus ISIS can be used for Simulation.
5. **PCB Designing Software:** PCB design software can be used to design layout of Printed Circuit Board. Possible PCB Designing software that can be used in this project are Eagle, EasyEDA, Fritzing.

### Debugger: It is a critical tool for testing. It goes through the code and eliminates bugs and errors, notifying places where they occur. In this project, we can perform BeagleBone Black remote Application Debugging with Eclipse or we can use GNU Debugger (GDB).

**Hardware Requirement:**

1. Beaglebone Black
2. Accelerometer Sensors
3. Motor Driver IC (L293D)
4. Wi-Fi Module (ESP 32)
5. Buzzer
6. Power Supply
7. GSM module
8. DC Motors
9. L293D driver motor IC.
10. Red Push Button

**Hardware Tools:**

**1. Digital Multimeter:** This device is used to measure current, voltage and resistance at different points in the circuit. Moreover, it is used to check connectivity between two points.

**2. Wires:** Different kind of wires can be used in an embedded project such as, jumper wires, BreadBoard wires, male-female wires etc.

**3. Soldering iron:** To connect components together, or to fix components on your circuit board, you need to solder them using soldering iron.

**4. De-soldering Gun:**Desolder means to remove the solder usually from a joint. It serves the opposite function of a soldering gun.

**5. Laptop:** Laptop is necessary tool and laptop must be equipped with required software tools.

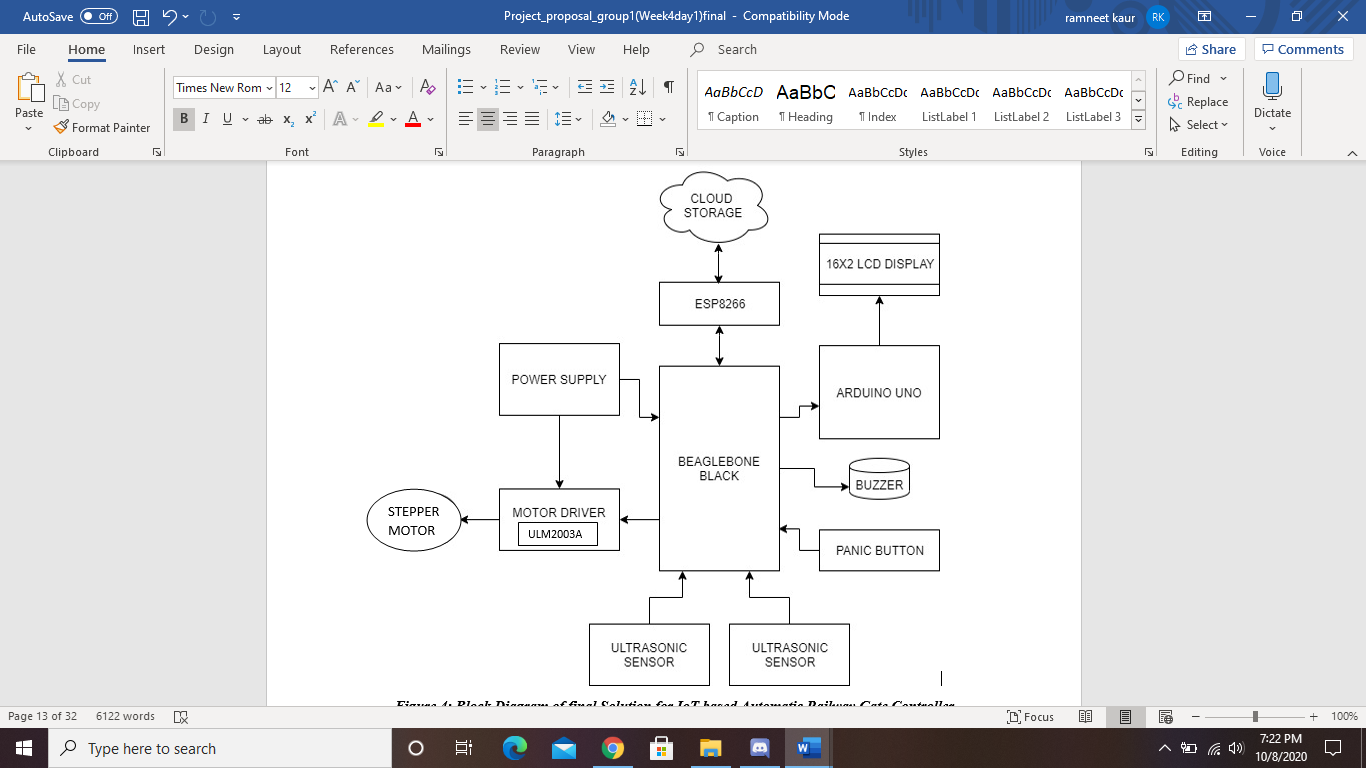
**6. Wire Cutter:**A cutter also known as wire cutter or clipper is used to cut jumper wires.

**7. Wire Stripper:**A wire stripper is used to remove insulation from wires without cutting them.

**References:**

* Balamurugan, C., Vijayshankarganth, P., Alagarraja, R., Subramanian, V., &Ragupathy, R. (2018). Automatic Railway Gate Control System Using 8051micro Controller. *International Journal of ChemTech Research*. doi:10.20902/ijctr.2018.110407
* Automatic Railway Gate Control System with High Speed Alerting System. (2017, December 25). Retrieved September 14, 2020, from <https://www.electronicshub.org/automatic-railway-gate-controller/>
* BeagleBoard.org. (n.d.). *BeagleBone Black*. Retrieved from <https://beagleboard.org/black>

**Solution 2:**

****

***Figure 3: Block Diagram of Solution 2 for IoT based Automatic Railway Gate Controller***

**Features of Solution 2:**

* Use of BeagleBone Black as master device which is 32-bit microprocessor with high processing speed.
* Use of IoT through Wi-Fi for cloud storage, such as, ThingSpeak.
* **Use of various peripherals such as touch screens, cameras, microphones and speakers, GPIOs, timers, GPS modules, Bluetooth, WiFi, and ADC/DACs?**
* Use of ESP8266 module which has Wi-Fi using IEEE802.11 protocol.
* Use of LCD to display messages in case of Emergency and status of the train.
* Use of Stepper motor interfaced with motor driver circuit ULN2003 for closing and opening of gates.
* Use of the Ultrasonic sensors to detect train on the railway track.
* Use of Panic Button that will be pressed in an emergency situation.
* Use of Buzzer to alert the users about upcoming trains and closure of gates**.**
* Use of Arduino Uno for interfacing LCD Display with BeagleBone Black.
* **Use of I2C, SPI, RS232/RS-485, IrDA infrared, JTAG, USB, Bluetooth, IEEE 802.11 WiFi, IEEE 802.3 Ethernet, CAN and GPS protocols and systems?**
* The Wi-Fi protocol used is IEEE802.11.
* The protocol used for communicating with cloud is MQTT.
* Parallel communication can be used between Arduino Uno and LCD Display.
* ULN2003 motor driver IC can be used to interface stepper motor with BeagleBone Black and it is connected to GPIO pins of BBB.
* Serial UART interfacing can be done between ESP 8266 and BeagleBone Black.
* GPIO pins of BeagleBone Black can be used to interface Buzzer.
* Panic Button is directly interfaced with GPIO pins of BBB.
* Interfacing between Ultrasonic sensor and BeagleBone Black can be done through GPIO pins.
* UART Protocol is used to interface Arduino Uno with BeagleBone Black.
* **Use of preemptive versus cooperative scheduler operation; tick rate and time slicing; critical code; fixed, dynamic and hybrid task priority allocation; application-specific considerations; power management tactics; semaphores, mutexes and queues; debugging strategies; performance estimation?**
* The system uses Real time concepts such as the pre-emptive scheduler operation wherein each task related to sensors and other peripherals will be assigned priorities for their functioning.
* The Panic button feature will act as an interrupt thus will have critical code for that section.
* Since the system uses pre-emptive scheduler, thus the tasks will be time-sliced else it will non pre-emptive in nature.
* Also, due to some conflicts of the resources may occur, thus we may employ the semaphores/ binary semaphores (i.e. mutexes) so that each task that executes will have to obtain the key before the execution.

**Software and Hardware Requirements for Solution 2:**

**Software Requirements:**

1. Debian Latest Image for the Beaglebone Black
2. Easy EDA/ Fritzing/ Proteus ISIS
3. ThingsSpeak
4. C Programming/ Python
5. Eclipse IDE
6. GCC Compiler or any cross compiler
7. Arduino IDE

**Software Tools:**

1. **Text editor:** To write source code in programming languages C and C++ and save this code as a text file, Nano, Vim text editors will be used in the project.
2. **Integrated Development Environment (IDE):** Eclipse IDE for programming in C language as  software that provides a set of necessary tools in one package. Arduino IDE to write program for Arduino Uno.
3. **Compiler:** A compiler is a tool for transforming the code into a low-level machine language code. GCC Compiler or any cross compiler such as Linaro can be used in this project.Other compiler that we can use in this project is ARM GCC Cross Compiler Toolchain.
4. **Simulation Software:** Simulation is the process of designing a model of a real system.Easy EDA/ Fritzing/Proteus ISIS can be used for Simulation.
5. **PCB Designing Software:** PCB design software can be used to design layout of Printed Circuit Board. Possible PCB Designing software that can be used in this project are Eagle, EasyEDA, Fritzing.

### Debugger: It is a critical tool for testing. It goes through the code and eliminates bugs and errors, notifying places where they occur. In this project, we can perform BeagleBone Black remote Application Debugging with Eclipse or we can use GNU Debugger (GDB).

**Hardware Requirements:**

1. BeagleBone Black
2. Ultrasonic Sensors (HC-SR04)
3. Motor Driver IC (ULN2003)
4. Wifi Module (ESP 8266)
5. Buzzer
6. Power Supply
7. LCD Display
8. Red Push Button
9. Stepper Motors
10. Arduino Uno

**Hardware Tools:**

**1. Digital Multimeter:** This device is used to measure current, voltage and resistance at different points in the circuit. Moreover, it is used to check connectivity between two points.

**2. Wires:** Different kind of wires can be used in an embedded project such as, jumper wires, BreadBoard wires, male-female wires etc.

**3. Soldering iron:** To connect components together, or to fix components on your circuit board, you need to solder them using soldering iron.

**4. De-soldering Gun:** Desolder means to remove the solder usually from a joint. It serves the opposite function of a soldering gun.

**5. Laptop:** Laptop is necessary tool and laptop must be equipped with required software tools.

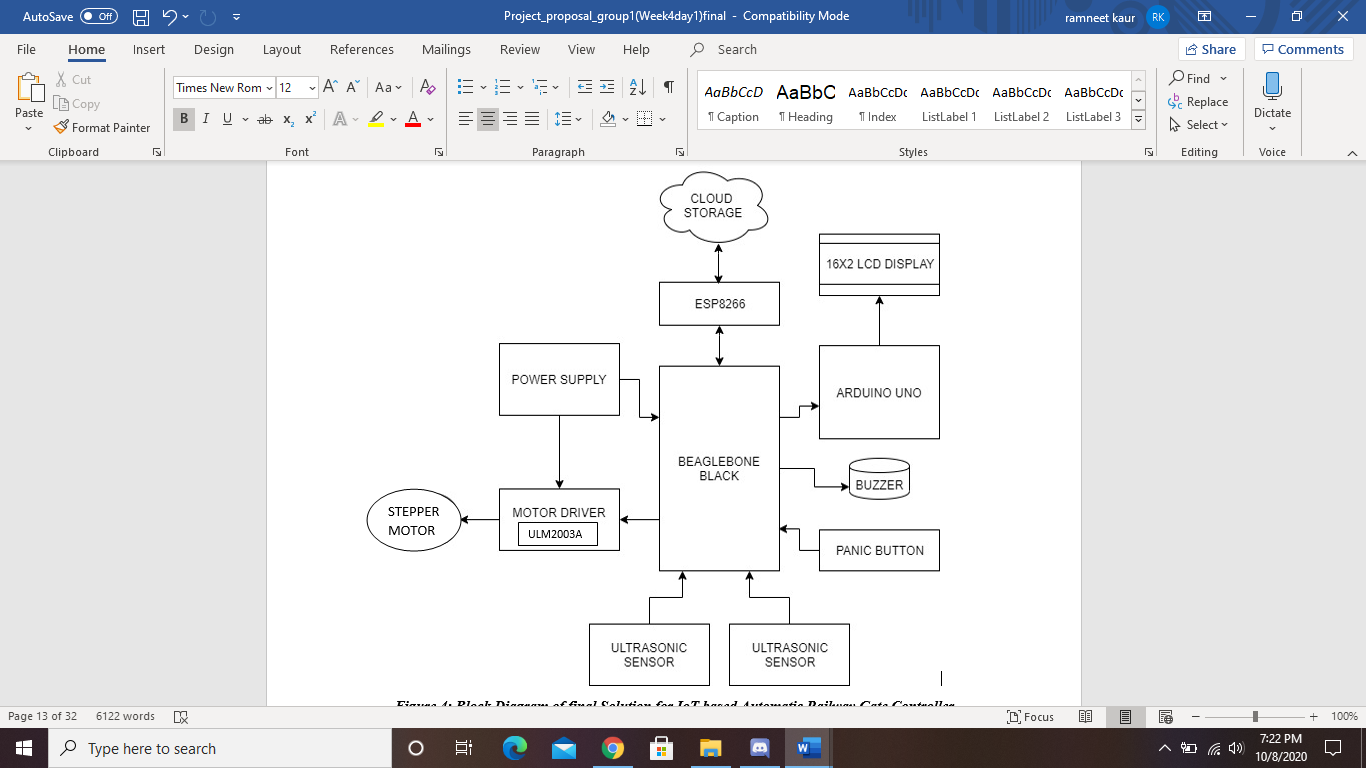
**6. Wire Cutter:** A cutter also known as wire cutter or clipper is used to cut jumper wires.

**7. Wire Stripper:** A wire stripper is used to remove insulation from wires without cutting them.

**References:**

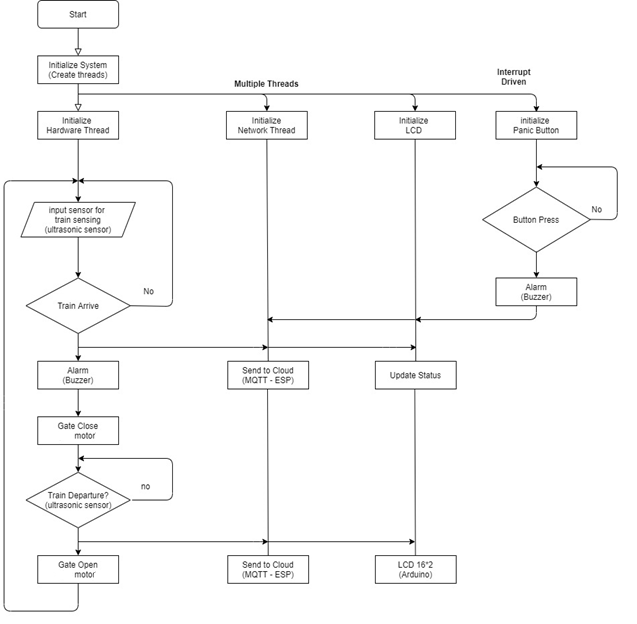
* Balamurugan, C., Vijayshankarganth, P., Alagarraja, R., Subramanian, V., &Ragupathy, R. (2018). Automatic Railway Gate Control System Using 8051micro Controller. *International Journal of ChemTech Research*. doi:10.20902/ijctr.2018.110407
* Automatic Railway Gate Control System with High Speed Alerting System. (2017, December 25). Retrieved September 14, 2020, from <https://www.electronicshub.org/automatic-railway-gate-controller/>
* BeagleBoard.org. (n.d.). *BeagleBone Black*. Retrieved from <https://beagleboard.org/black>
* ＜Accelerometer sensors＞. (n.d.). Retrieved September 21, 2020, from <https://www.rohm.com/electronics-basics/sensor/accelerometer-sensor>

**Final Solution:**

****

***Figure 4: Block Diagram of final Solution for IoT based Automatic Railway Gate Controller***

**Flowchart:**

****

**Features of Final Solution:**

* Use of BeagleBone Black as master device which is 32-bit microprocessor with high processing speed.
* Use of IoT through Wi-Fi for cloud storage, such as, ThingSpeak.
* **Use of various peripherals such as touch screens, cameras, microphones and speakers, GPIOs, timers, GPS modules, Bluetooth, WiFi, and ADC/DACs?**
* Use of ESP8266 module which has Wi-Fi using IEEE802.11 protocol.
* Use of LCD to display messages in case of Emergency and status of the train.
* Use of Stepper motor interfaced with motor driver circuit ULN2003 for closing and opening of gates.
* Use of the Ultrasonic sensors to detect train on the railway track.
* Use of Panic Button that will be pressed in an emergency situation.
* Use of Buzzer to alert the users about upcoming trains and closure of gates**.**
* Use of Arduino Uno for interfacing LCD Display with BeagleBone Black.
* **Use of I2C, SPI, RS232/RS-485, IrDA infrared, JTAG, USB, Bluetooth, IEEE 802.11 WiFi, IEEE 802.3 Ethernet, CAN and GPS protocols and systems?**
* The Wi-Fi protocol used is IEEE802.11.
* The protocol used for communicating with cloud is MQTT.
* Parallel communication can be used between Arduino Uno and LCD Display.
* ULN2003 motor driver IC can be used to interface stepper motor with BeagleBone Black and it is connected to GPIO pins of BBB.
* Serial UART interfacing can be done between ESP 8266 and BeagleBone Black.
* GPIO pins of BeagleBone Black can be used to interface Buzzer.
* Panic Button is directly interfaced with GPIO pins of BBB.
* Interfacing between Ultrasonic sensor and BeagleBone Black can be done through GPIO pins.
* UART Protocol is used to interface Arduino Uno with BeagleBone Black.
* **Use of preemptive versus cooperative scheduler operation; tick rate and time slicing; critical code; fixed, dynamic and hybrid task priority allocation; application-specific considerations; power management tactics; semaphores, mutexes and queues; debugging strategies; performance estimation?**
* The system uses Real time concepts such as the pre-emptive scheduler operation wherein each task related to sensors and other peripherals will be assigned priorities for their functioning.
* The Panic button feature will act as an interrupt thus will have critical code for that section.
* Since the system uses pre-emptive scheduler, thus the tasks will be time-sliced else it will non pre-emptive in nature.
* Also, due to some conflicts of the resources may occur, thus we may employ the semaphores/ binary semaphores (i.e. mutexes) so that each task that executes will have to obtain the key before the execution.

**Software and Hardware Requirements for Final Solution:**

**Software Requirements:**

1. Debian Latest Image for the BeagleBone Black
2. Easy EDA/ Fritzing/Proteus ISIS
3. ThingsSpeak
4. C Programming/ C++ Programming
5. Eclipse IDE
6. GCC Compiler or any cross compiler
7. Arduino IDE

**Software Tools:**

1. **Text editor:** To write source code in programming languages C and C++ and save this code as a text file, Nano, Vim text editors will be used in the project.
2. **Integrated Development Environment (IDE):** Eclipse IDE for programming in C language as  software that provides a set of necessary tools in one package. Arduino IDE to write program for Arduino Uno.
3. **Compiler:** A compiler is a tool for transforming the code into a low-level machine language code. GCC Compiler or any cross compiler such as Linaro can be used in this project.Other compiler that we can use in this project is ARM GCC Cross Compiler Toolchain.
4. **Simulation Software:** Simulation is the process of designing a model of a real system. Easy EDA/ Fritzing /Proteus ISIS can be used for Simulation.
5. **PCB Designing Software:** PCB design software can be used to design layout of Printed Circuit Board. Possible PCB Designing software that can be used in this project are Eagle, EasyEDA, Fritzing.

### Debugger: It is a critical tool for testing. It goes through the code and eliminates bugs and errors, notifying places where they occur. In this project, we can perform BeagleBone Black remote Application Debugging with Eclipse or we can use GNU Debugger (GDB).

**Hardware Requirements:**

1. BeagleBone Black
2. Ultrasonic Sensors (HC-SR04)
3. Motor Driver IC ( ULN2003)
4. Wifi Module (ESP 8266)
5. Buzzer
6. Power Supply
7. LCD Display
8. Push Button
9. Stepper Motor
10. Arduino Uno

**Hardware Tools:**

**1. Digital Multimeter:** This device is used to measure current, voltage and resistance at different points in the circuit. Moreover, it is used to check connectivity between two points.

**2. Wires:** Different kind of wires can be used in an embedded project such as, jumper wires, Breadboard wires, male-female wires etc.

**3. Soldering iron:** To connect components together, or to fix components on your circuit board, you need to solder them using soldering iron.

**4. De-soldering Gun:** Desolder means to remove the solder usually from a joint. It serves the opposite function of a soldering gun.

**5. Laptop:** Laptop is necessary tool and laptop must be equipped with required software tools.

**6. Wire Cutter:** A cutter also known as wire cutter or clipper is used to cut jumper wires.

**7. Wire Stripper:** A wire stripper is used to remove insulation from wires without cutting them.

**Cost Estimation Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. no.** | **Components** | **Part number** | **Quantity** | **Price**  **(including taxes in $)** | **References** |
| 1. 1. | BeagleBone Black | BBB01-SC-505-ND | 1 | $ 102 | <https://www.digikey.ca/en/products/detail/ghi-electronics,-llc/BBB01-SC-505/6210999?utm_adgroup=Evaluation%20Boards%20-%20Embedded%20-%20MCU%2C%20DSP&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Development%20Boards%2C%20Kits%2C%20Programmers&utm_term=&productid=6210999&gclid=Cj0KCQjwtsv7BRCmARIsANu-CQcVuxoFiuOAY7eOQmMO3avYaczICpFGrhki1ZwkFZruwoePrbEY0bkaAlYFEALw_wcB> |
| 1. 2. | Ultrasonic Sensors (HC-SR04) | CA-MW-Sensor-HC-SR04-2PCS | 2 | $ 10 | <https://www.amazon.ca/gp/product/B07XBY7DX3/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1> |
| 1. 4. | Wifi Module  (ESP 8266) | NodeMCU ESP-12E | 1 | $ 11 | <https://www.amazon.ca/gp/product/B07PR9T5R5/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1> |
| 1. 5. | Buzzer | 2056-MB14-12V-ND | 1 | $ 3 | <https://www.digikey.ca/en/products/detail/visaton-gmbh-co-kg/MB-14-12-V/10650344> |
| 1. 6. | Power Supply  (wall adapter) | 237-1385-ND | 1 | $ 16 | <https://www.digikey.ca/en/products/detail/triad-magnetics/WSU050-2000/3094911?s=N4IgTCBcDa4MwHYC0BGOAOArEgcgERAF0BfIA> |
| 1. 7. | LCD Display | 635-1205-ND | 1 | $ 16 | <https://www.digikey.ca/en/products/detail/matrix-orbital/MOP-AL162A-BYFY-25J-3IN/9602838> |
| 1. 8. | Push Button | 450-1653-ND | 1 | $ 0.50 | <https://www.digikey.ca/en/products/detail/te-connectivity-alcoswitch-switches/1-1825910-4/1632539?s=N4IgTCBcDaIIwFo4A4wFYCccAMCAsIAugL5A> |
| 1. 9. | 5V 4-Phase Stepper Step Motor + Driver Board ULN2003 | 28BYJ-48,  ULN2003 | 1 | $ 20 | <https://www.amazon.ca/Yoochin-4-Phase-Stepper-Machinery-Raspberry/dp/B076DC1B2W/ref=sr_1_2_sspa?dchild=1&keywords=ULN2003&qid=1602186914&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFJMkwwR1lXTENEMU0mZW5jcnlwdGVkSWQ9QTA2NTcwMTUxVU5BMFdLUUhTRVgzJmVuY3J5cHRlZEFkSWQ9QTA0NjMxMzIzR1FNRjhFVTQ2Mko2JndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ>== |
| 1. 10. | Arduino Uno | 1050-1024-ND | 1 | $ 35 | <https://www.digikey.ca/en/products/detail/A000066/1050-1024-ND/2784006?itemSeq=340030105> |
| 1. 11. | Basic Soldering kit with Digital Multimeter and tools | Soldering Iron Kit | 1 | $ 70 | [https://www.amazon.ca/Soldering-Iron-Kit-Rarlight-Multimeter/dp/B07NVL3BX6/ref=sr\_1\_11?crid=DP6AE3NHT5BX&dchild=1&keywords=soldering+kit+with+multimeter&qid=1601389466&sprefix=soldering+kit+%2Caps%2C192&sr=8-11](https://www.google.com/url?q=https://www.amazon.ca/Soldering-Iron-Kit-Rarlight-Multimeter/dp/B07NVL3BX6/ref%3Dsr_1_11?crid%3DDP6AE3NHT5BX%26dchild%3D1%26keywords%3Dsoldering%2Bkit%2Bwith%2Bmultimeter%26qid%3D1601389466%26sprefix%3Dsoldering%2Bkit%2B%252Caps%252C192%26sr%3D8-11&sa=D&source=hangouts&ust=1601476579921000&usg=AFQjCNGnS9djogcySemUWMU4l96FC1O3VA) |
| 1. 12. | Toy Train |  | 1 | $ 56 | <https://www.amazon.ca/Haktoys-Classical-Simulation-Locomotive-Authentic/dp/B015975ER2/ref=sr_1_7?dchild=1&keywords=toy+train&qid=1601329735&sr=8-7> |
| 1. 13. | PCB Designing |  | 1 | $ 100 | Approximate |
| 1. 14. | Zero PCB | SBBTH3030-1-ND | 1 | $ 9.5 | <https://www.digikey.com/product-detail/en/chip-quik-inc/SBBTH3030-1/SBBTH3030-1-ND/5978259> |
| 1. 15. | Connecting Cables | EL-CP-004 | 1 | $13.5 | <https://www.amazon.ca/Elegoo-120pcs-Multicolored-Breadboard-arduino/dp/B01EV70C78/ref=sr_1_16?dchild=1&keywords=connecting+wire+electronics&qid=1601334485&sr=8-16> |
| **Total Price** | |  | **$ 462.5** | | |

**Milestones (Deliverables and Time Schedule):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. no.** | **Task Name** | **Start Date** | **End Date** | **Person in-charge** |
|  | Project Proposal | Sep 18, 2020 | Oct 9, 2020 |  |
|  | Finalizing Hardware requirements and component order | Oct 9, 2020 | Oct 13, 2020 | Ramneet Kaur |
|  | Schematic Designing | Oct 13, 2020 | Oct 19, 2020 | Navjot Saini |
|  | Testing each hardware parts | Oct 20, 2020 | Oct 25, 2020 | Ramneet Kaur |
|  | Interfacing Panic Button and Buzzer with BBB | Oct 26, 2020 | Oct 31, 2020 | Navjot Saini |
|  | Interfacing Ultrasonic Sensors with BeagleBone Black | Nov 1, 2020 | Nov 5, 2020 | Ramneet Kaur |
|  | Interfacing stepper motor with BeagleBone Black | Nov 6, 2020 | Nov 11, 2020 | Navjot Saini |
|  | Interfacing ESP 8266 with BBB | Nov 12, 2020 | Nov 19, 2020 | Ramneet kaur |
|  | Interfacing Arduino and LCD with BBB. | Nov 20, 2020 | Nov 26, 2020 | Navjot Saini |
|  | Interfacing ThingSpeak with ESP8266 | Nov 27, 2020 | Dec 1, 2020 | Ramneet Kaur |
|  | PCB Design | Dec 28, 2020 | Dec 3, 2020 | Navjot Saini |
|  | Final project Testing | Dec 4, 2020 | Dec 10, 2020 |  |
|  | Final Report presentation | Dec 11, 2020 | Dec 15, 2020 |  |
|  | Final Project Demo | Dec 18, 2020 |  |  |

**Communication Standards to be used in this Project:**

In this project we will majorly use 2 protocols for communicating with other devices which are

1. UART: beagle bone black to Arduino Uno, beagle bone black to ESP
2. Parallel Communication: Arduino Uno to LCD.
3. GPIO connections: Ultrasonic sensor to BBB, Panic Button to BBB, Buzzer to BBB, Motor to BBB.

# UART Protocol:

UART or Universal Asynchronous Receiver Transmitter protocol is widely used protocol to communicate between two devices in the embedded systems. UART is one of the most common and easy to use protocols. It is used in multiple applications where we want two devices to communicate with each other without anyone being a master or slave. UART is used in NCP (network co-processor) based applications like Bluetooth modules, GPS modules, GPRS modems etc.

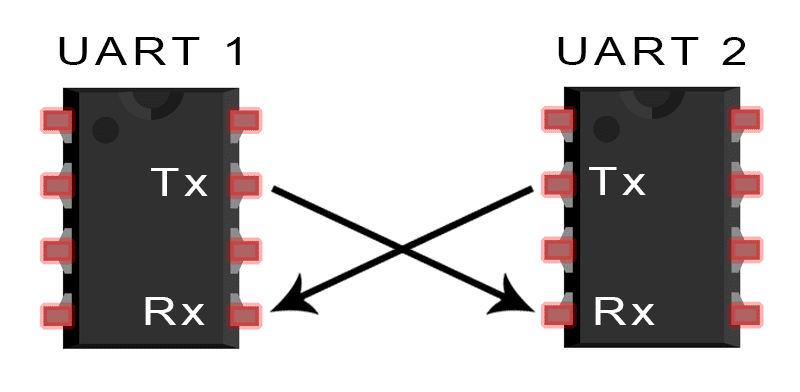
Most of the microcontrollers have dedicated hardware unit for UART build in it to their architecture. The reason for integrating it inside the controller is that it requires only two pins to communicate with other devices. Two pins can be used to transfer data between two devices. Tx Pin or transmit pin of one device is connected to Rx pin i.e. receive pin of another device.

## Some of the main features and advantages of UART

* Its asynchronous serial protocol i.e. it does not require clock signals
* Its bidirectional, both reception and transmission can happen in parallel.
* It has only two data lines i.e. TX (to transmit data) and RX (to receive data)
* UART can also handle synchronization between devices using extra pins

## Disadvantages of UART protocol

* Size of data frame is limited to only 9 bits
* Cannot use multiple devices in the system
* Speed of the communication is low as compared to parallel communication.



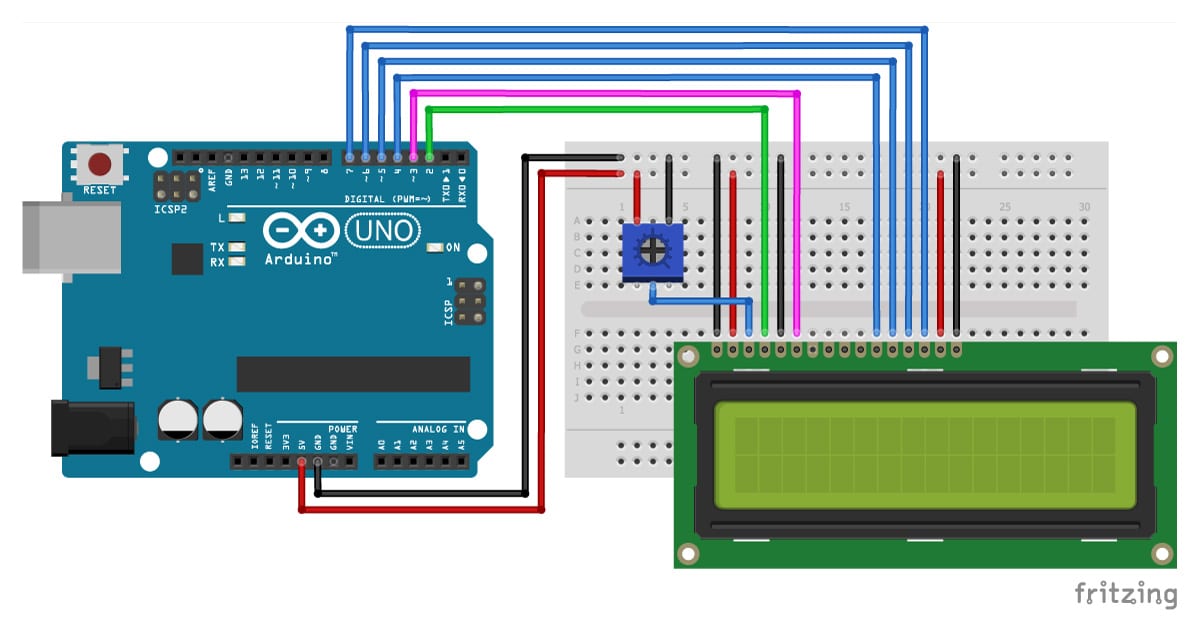
***Figure 5: UART Communication Protocol***

**Modules interfaced through UART communication Protocol in this project:**

* We are also using NCP based approach i.e. we will send messages to ESP module and will share network related work to ESP module. ESP8266 is an SOC with support for Wi-Fi Protocol, it will connect to a preconfigured Wi-Fi SSID and then connect with MQTT to send and receive messages.
* Arduino Uno will be interfaced through UART protocol with BeagleBone Black as it requires only TX and RX pins of BBB to implement serial communication with Arduino.

# Parallel Communication Protocol:

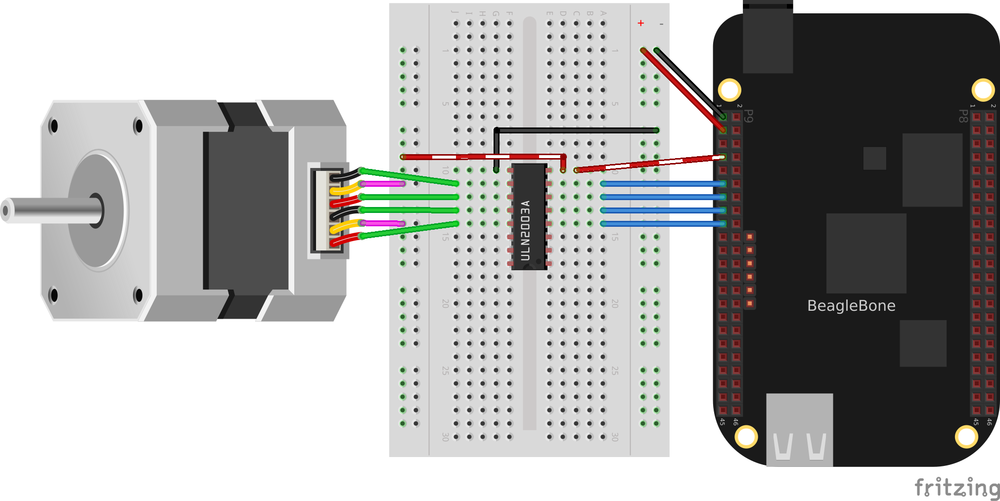
In parallel communication protocol,multiple parallel links are used to transmit data of each bit simultaneously. It is used in places which don’t require being cost effective but needs higher speeds of data transfer. The cost of the system will be higher as the number of pins required is more, and each bit will carry each bit. Parallel communication is half duplex protocol, i.e. data can be transferred in only one direction and it is highly efficient in short communication distances. In our project we will use parallel communication for following Modules:

* Between LCD display and Arduino board, we can either use the LCD in 4 pin mode or 8 pin mode, but in our project, we will use 8 pin mode.

***Figure 6: Parallel communication between Arduino Uno and LCD display***

**Direct GPIO Connections:**

* General purpose pins of BBB will be used for interfacing Stepper motor and Beaglebone black. ULN2003 motor driver IC is used to drive speed and direction of DC motor.



***Figure 7: GPIO interfacing between Stepper motor and BBB***

* Interfacing between Ultrasonic sensor and BeagleBone Black can be done directly over GPIO pins.
* Similarly, Push button for emergency situations and buzzer can also be directly interfaced on BBB using GPIOs.

# MQTT Protocol:

MQTT stands for Message Queue Telemetry Transport is a lightweight messaging protocol designed for Machine to machine telemetry in resource constrained environments like embedded systems, it is one of the main protocols for IOT. MQTT is publish publish/subscribe based model i.e. it is event driven and enabled messages to be pushed to clients.

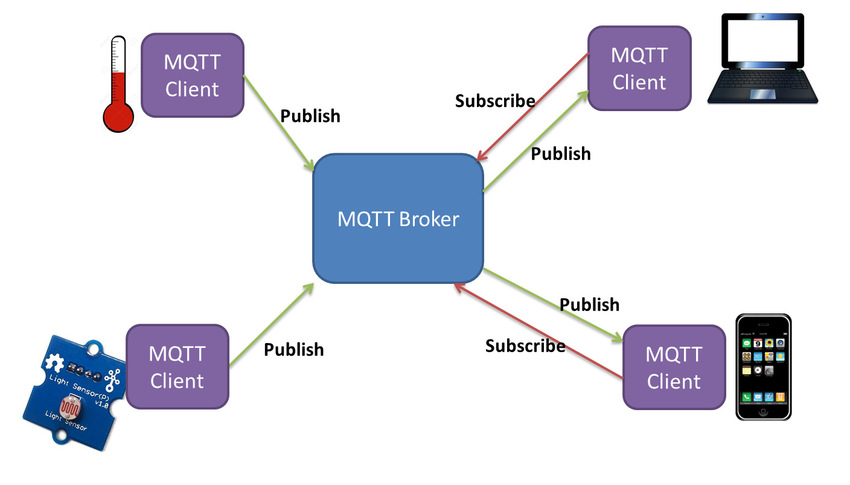
MQTT has wide range of applications from sensor networks to automotive industry for connected cars. It can be used in smart healthcare devices to update the statistics of patient in real time to the doctor. In this project, The ThingSpeak IoT platform will receive updates from channel feeds via the ThingSpeak MQTT broker. MQTT is a publish/subscribe communication protocol that uses TCP/IP sockets.

## Advantages of MQTT protocol

* Lightweight and efficient, making it ideal for use in resource constrained devices like embedded systems.
* It allows for bidirectional communication between server and devices, with support for broadcast messages to a group.
* Easily scalable to huge number of devices.
* MQTT supports client authentication protocols and message encryption using TLS.

## Basic Concepts

The two most important parts of MQTT protocol are the MQTT Broker and MQTT client. An MQTT client publishes a message to the broker on a specific topic. Clients can also subscribe to specifics topics to the broker. Broker is responsible for sending messages between to the rightful receivers. Once a broker receives a message on a specific topic, it should send that message to other clients what have subscribed to the same topic.



***Figure 8: MQTT communication Protocol***

**Coding Standards:**

Coding Standard was designed specifically to reduce the number of programming defects in embedded software. By following this coding standard, firmware developers not only reduce hazards to users and time spent in the debugging stage of their projects but also improve the maintainability and portability of their software. Embedded software developers focus primarily on C code which is primary language of firmware designers.

**MISRA-C coding Standards:**

MISRA C is a set of software development guidelines for the C programming language developed by the Motor Industry Software Reliability Association (MISRA). The guidelines aim to facilitate code safety, security, portability, and reliability in embedded systems. MISRA has evolved into a widely accepted model for best practices by leading developers in sectors including automotive, aerospace, telecom, medical devices, defense, railway, and others. In all, MISRA C has 127 rules. Of these, 93 are required and the remaining 34 are advisory. MISRA standards are used to ensure that code is safe, secure, reliable and portable.

**MISRA-C coding standards rules to be followed:**

* All code shall conform to ISO 9899 standard C, with no extensions permitted. (Required Rule)
* Comments shall not be nested. (Required)
* Sections of code should not be ‘commented out’. (Advisory)
* The type char shall always be declared as unsigned char or signed char. (Required)
* The basic types of char, int, short, long, float and double should not be used, but specific-length equivalents should be typedef’d for the specific compiler, and these type names used in the code. (Advisory)
* The typedef names shall not be reused. (Required)
* All object and function identifiers shall be declared before use. (Required)

**MISRA C++ coding Standards:**

C++ programming language can be used for embedded applications, high integrity and safety related systems. Rules that will be considered while coding in C++ language are:

* The project shall not contain unused variables. (Required)
* Every defined function must be called at least once. (Required)
* There should not be unused parameters. (Required)
* If the function generates error information, then error information should be tested. (Required)
* Section of code should not be commented out. (Advisory)
* The typedef name shall be unique identifier. (Required)
* The class or enum name shall be unique identifier. (Required)

**, Environmental and legal ramifications in the Project:**

In this section, we have discussed about possible environmental, legal, and ethical ramifications in the project design. The potential applications of the project will be at the railway crossing gates, which are currently managed manually. It will replace the manual intervention, with automatic gates opening and closing operations, when a train is about to arrive.

**Ethical Issues:**

* **Safety and sustainability** is major ethical issues that play a dominant role in many design processes. Our design for IoT based Automatic railway gate controller should be safe enough to be deployed on the Railway Track. While developing engineering design, safety of the product is of prime important. In our project, the product should be safe for incoming traffic crossing the railway line and passengers of the train. However, any defects in the design of the product can cause fatal accident in real life scenario.
* **Sustainability** refers to sustainable designs such that it does not utilize energy and resources that affect future generation needs. Moreover, sustainable designs consider environmental, social and economic constraints. This project does not cause any issue related to sustainability as resources used in the project are not depleted. This project reduces the power consumption as the controllers will go into standby mode when it is idle and will only wakeup on interrupts from either train detection or from a panic button.
* **Security** is another issue that can be identified in the Internet of Things (IoT). If not well-protected, the sensor network can become a weak point through which unauthorized access can become a threat to the secrecy of data. In this project, MQTT protocol allows for authentication which enables a two-way handshake. Even though security was not designed into the protocol, there are some security safeguards it provides. However, MQTT protocol is not referred to the networks which require high end security such as for military applications. However, for this project, MQTT protocol provides sufficient message encryption through Transport layer Security (TLS) layer.

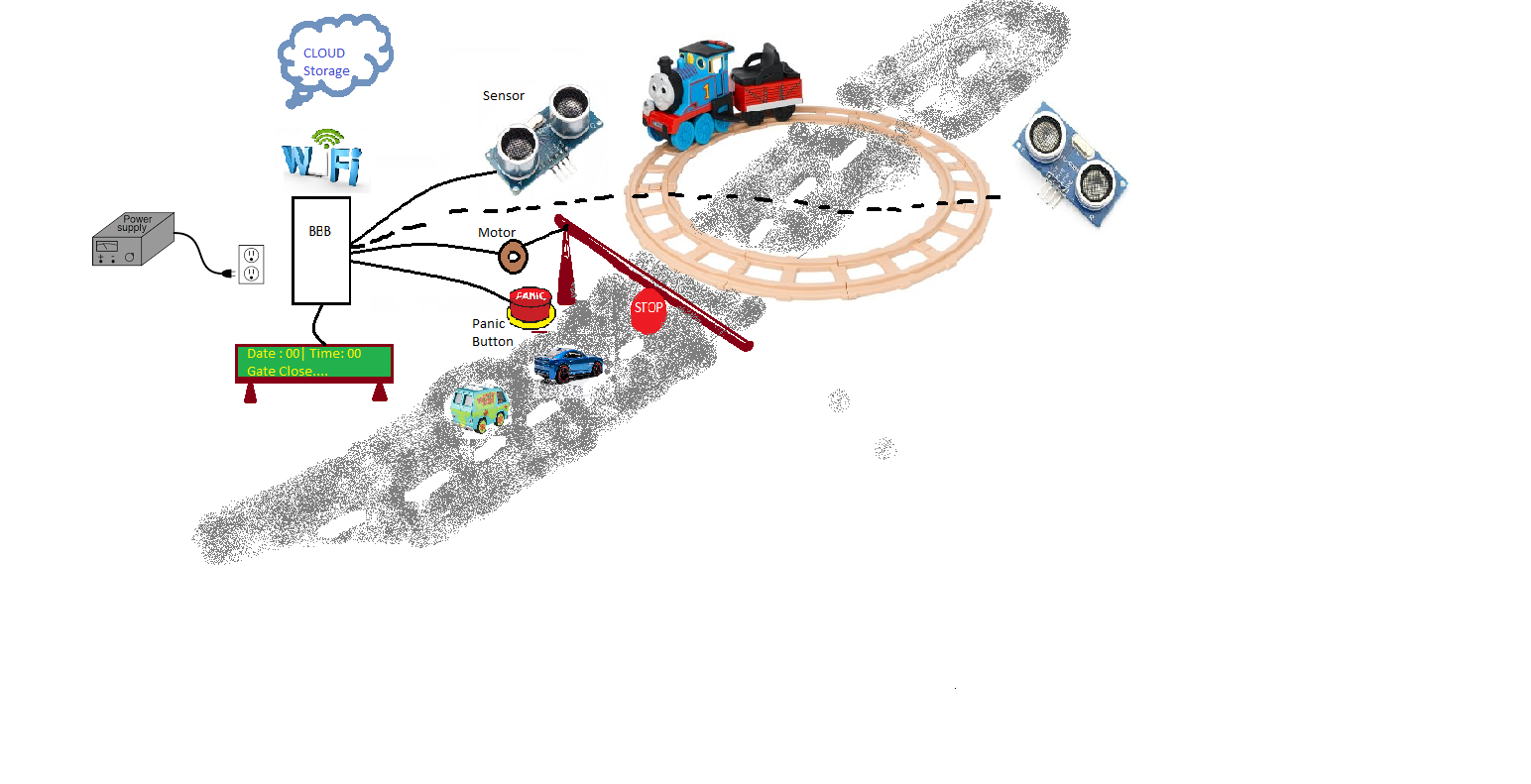
**Environmental Issues:**

* The use of Wi-Fi devices is becoming more growing concern about the possible long-term health effects of the exposure to the **Electromagnetic Field radiation**, or **EMF radiation** emitted from our mobile devices, cell towers, power lines, and wireless equipment. The emission of EMF radiation can cause harmful effect on plants, animals and humans in long term.
* Ultrasonic sensors emit **ultrasonic waves** that cannot be heard by humans, at high decibels it can still cause direct damage to human ears. High frequency sound and ultrasound with sufficient intensity can cause hearing loss, headache, fatigue, dizziness and nausea in long term.
* Electronics/embedded components contain material like mercury, lead, copper, aluminum, flammable lithium ion batteries etc. are not decompose easily and they have expensive recycling.

**Legal Issues:**

* This project uses some libraries like the MQTT library, Arduino and Wi-Fi library, etc. all these libraries are under MIT license or Open source so they can be used in our project without paying any fees.
* Also, all the code developed in this project will be shared with the community and licensed under MIT license. In this project, we are using Wi-Fi protocol which works at 2.4Ghz in the free spectrum. Hence all the technologies used in the project are either unlicensed or under free to use license.
* It is our legal responsibility to ensure that all the work done in the project is authentic and unique.

**Engineering Drawing of the Project:**



**Figure 9: Engineering Drawing of the Project**

**Bill of Materials:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item no.** | **Quantity** | | **Part Name** | **Part number** | **Manufacturer** | **Description** | **BOM Notes** |
| 1. | | 1 | BeagleBone Black | BBB01-SC-505-ND | GHI Electronics, LLC | AM3358BZCZ BeagleBone Black Rev C series ARM® Cortex®-A8 MPU Embedded Evaluation Board | Single core Master/Controlling Device |
| 2. | | 2 | Ultrasonic Sonar Distance Sensor (HC-SR04) | CA-MW-Sensor-HC-SR04-2PCS | HUABAN | ultrasonic sensor sonar distance | Senses the arrival and departure of the train |
| 3. | | 1 | Wi-Fi Module  (ESP 8266) | NodeMCU ESP-12E | KeeYees | Support wireless 802.11b/g/n standard, Built-in TCP/IP protocol stack, supports multiple TCP Client connections (5MAX) | Provide any microcontroller with access to Wi-Fi network |
| 4. | | 1 | Buzzer | 2056-MB14-12V-ND | Visaton GmbH & Co. KG | Buzzers | 1.Alarms on the arrival of the train  2. Also signal the emergency situation |
| 5. | | 1 | Power Supply  (wall adapter) | 237-1385-ND | Triad Magnetics | AC/DC wall mount adapter 5V 10W | Provide power to devices/ Beaglebone Black |
| 6. | | 1 | 16X2 LCD Display | 635-1205-ND | Matrix Orbital | Character LCD Display Module 32DIG 16X2 | Show output as text messages |
| 7. | | 1 | Push Button | 450-1653-ND | TE Connectivity ALCOSWITCH Switches | Tactile Switch SPST-NO 0.05A 24V | Panic button pressed when Emergency occurs |
| 8. | | 1 | 5V 4-Phase Stepper Step Motor + Driver Board ULN2003 | 28BYJ-48  ULN2003 | Yoochin | 5V 4 Phase DC Gear Stepper Motor + ULN2003 Driver Board UNO MEGA R3 DIY Starter Kit | Control status of gate (open or close) |
| 9. | | 1 | Arduino Uno | 1050-1024-ND | Arduino | ATmega328P Arduino Uno R3 series AVR MCU 8-Bit Embedded Evaluation Board | Used for Interfacing Between LCD Display and Beaglebone Black |
| 10. | | 1 | Soldering Iron Kit with Multimeter | Soldering Iron Kit | SREMTCH | Soldering Iron Kit with ON/Off Switch, Rarlight 60W 110V Adjustable Temperature Welding Tool with Digital Multimeter, Soldering Tips, Desoldering Pump, Solder Wire, Tweezers, Stand, Wire Stripper Cutter | Contains basic Hardware tools required for the assembly of the Project |
| 11. | | 1 | Toy Train | - | Haktoys | Haktoys Railway King Classical Freight Train Set Battery Operated Ready to Play Simulation Steam Locomotive Playset with Smoke, Lights and Authentic Train Sound | Dummy Train |
| 13. | | 1 | Zero PCB | SBBTH3030-1-ND | Chip Quik Inc. | Breadboard, General Purpose Plated Through Hole (PTH) Pad Per Hole (Round) 0.100" (2.54mm) | Used as base to mount components for DIY Electronics Project |
| 14. | | 1 | Connecting Cables | EL-CP-004 | Elegoo | Elegoo 120pcs Multicolored Dupont Wire 40pin Male to Female, 40pin Male to Male, 40pin Female to Female Breadboard Jumper Wires Ribbon Cables Kit for arduino | Used for making connections on Breadboard |

**Edit-Test-Debug Cycle implementation phase of the development process:**

**1. A Target System (Used):** The primary mode of development for real-time embedded software has been the "host/target" approach. In this approach, the software for the application is developed on one computer system to be executed on another.

The target used in our project is **Beaglebone black.** The program file can be edited in bbb terminal in “vim ” mode using vim command or using “Nano”.

**VIM** is a text editor of bbb. Vim offers multi-level [undo](https://www.computerhope.com/jargon/u/undo.htm), multiple windows and buffers, [syntax](https://www.computerhope.com/jargon/s/syntax.htm) highlighting, [command line](https://www.computerhope.com/jargon/c/commandi.htm) editing, [file name](https://www.computerhope.com/jargon/f/filename.htm) completion, a complete help system, visual selection, and others. To make changes to program we need to enter “insert mode” by typing “i”.

**Nano** is a simple command-line text editor included in most Linux installations. With a simple easy to use interface. Nano offers searching, replacing, editing file etc.

**GDB, the**[**GNU Project Debugger**](http://linux.die.net/man/1/gdb) is a debugging tool provided with the GNU Compiler Collection (GCC). GDB allows you to stop and start a running program, examine its functioning, and make change which can be used to develop, install, and debug embedded applications.

**2. An Emulator (Not Used):**  Emulator enables the host system to run software, tools, peripheral devices and other components which are designed for the guest system. In this project, we are not using emulator to emulate our hardware and software.

**3. In-circuit Emulation (Not used):** In-circuit emulation (ICE) is the use of a hardware device or in-circuit emulator used to debug the software of an embedded system. In this project, we are not using In-circuit emulation.

**4. Joint Test Action Group (Not used):** JTAG is an industry standard for verifying designs and testing printed circuit boards after manufacture. In this project, we are not using JTAG.

**5. Background debug mode (BDM) interface (Not used):** Background debug mode (BDM) interface is an electronic interface that allows debugging of embedded systems. Specifically, it provides in-circuit debugging functionality in microcontrollers. In this project, we are not using BDM.

**6. A Simulator (Not used):** A simulator is designed to create an environment that contains all of the software variables and configurations that will exist in an application’s actual production environment. As simulators create only software environments, they can be implemented using high-level programming languages. In this project, we are not using simulator.

**7. Integrated Development Environment (Used):** An Integrated Development Environment (IDE) is software that assists programmers in developing software. IDEs normally consist of a source code editor, a compiler, a linker/locater and usually a debugger. Integrated Development Environment that will be used in this project is Eclipse IDE and Arduino IDE.

* **Use of Eclipse IDE:**

Eclipse IDE is one of the most widely used development environment for C and C++ applications. It is an Open source IDE for code editing, debugging, uploading and running applications from inside the development environment. It also has a built in Makefile generator, that can automatically generate and update Make-files in the background. Eclipse also has built-in support for remote debugging i.e. it can connect to remote systems using SSH and displays the files in the IDE, as if we are working on the same system. In our project we will use this feature of eclipse for our development on Beaglebone black. We need to install GDB server on the beaglebone to enable debugging of C/C++ code. After installing eclipse on our desktop, we just need to use the remote explorer plugin of the IDE to connect with BBB and then we can create, edit and debug projects like we do in our ubuntu machine.

* **Use of Arduino IDE:**

Arduino IDE is an open source development environment for compatible microcontrollers and development boards. Users can edit, compile and upload the code to their microcontrollers directly from the IDE. The IDE can be easily installed in Windows, Linux and MAC supporting devices. It has support for multiple microcontrollers like, Arduino UNO, Arduino Mega, ESP32, ESP8266 etc. Arduino also has a built in Serial Monitor, which can used to read data from serial port of the device. One of the main features of IDE is that it has built-in integration for adding hundreds of libraries. Users can install any library from these libraries and directly use it in the project.  
In our project we will use Arduino IDE for two modules  
**1)     Arduino UNO-LCD interfacing:**

In our project, Arduino Uno acts as a mediator between Beaglebone black and LCD interface. Arduino needs to read commands from UART interface form BBB and then display the information on the LCD using Parallel interface. We will use Arduino Library for LCD interfacing and Arduino-Serial library for UART communication with Beaglebone.  
2)     **ESP8266:**

This board is also compatible with Arduino IDE and its framework. As ESP has onboard WIFI support, in our project we will use it as a Network co-processor. Beaglebone will send commands to ESP using the UART interface to communicate. We will use ESP Wi-Fi Arduino library to connect with Wifi, PUBSUB client library for MQTT connection, NTP library for time syncing and Serial library for UART interfacing.

**References**

* Balamurugan, C., Vijayshankarganth, P., Alagarraja, R., Subramanian, V., &Ragupathy, R. (2018). Automatic Railway Gate Control System Using 8051micro Controller. *International Journal of ChemTech Research*. doi:10.20902/ijctr.2018.110407
* Automatic Railway Gate Control System with High Speed Alerting System. (2017, December 25). Retrieved September 14, 2020, from <https://www.electronicshub.org/automatic-railway-gate-controller/>
* BeagleBoard.org. (n.d.). *BeagleBone Black*. Retrieved from <https://beagleboard.org/black>
* Molloy, D. *Exploring BeagleBone, Second Edition.* Wiley.
* Sparkfun. (n.d.). *Ultrasonic Distance Sensor - HC-SR04.*Retrieved from <https://www.sparkfun.com/products/15569>
* ProgrammerSought. (n.d.). *Beaglebone Black – Smart Home Control System LAS - Connect to ESP8266 with UART (ESP-01 version)*. Retrieved from <https://www.programmersought.com/article/53081679253/>
* Sparkfun. (n.d.). *BeagleBone Black - Rev C*. Retrieved from <https://www.sparkfun.com/products/12857>
* Circuits Today. (n.d.). *Interfacing LCD to Arduino – Display Text and Characters on LCD Screen using Arduino*. Retrieved from <https://www.circuitstoday.com/interfacing-lcd-to-arduino>
* *Arduino*. (n.d.).DigiKey Electronics Retrieved from: <https://www.digikey.ca/en/products/detail/A000066/1050-1024-ND/2784006?itemSeq=340030105>
* *Haktoys Railway King Classical Freight Train Set Battery Operated.* (n.d.). Retrieved from Amazon.ca: <https://www.amazon.ca/Haktoys-Classical-Simulation-Locomotive-Authentic/dp/B015975ER2/ref=sr_1_7?dchild=1&keywords=toy+train&qid=1601329735&sr=8-7>
* MQTT. (n.d.). *MQTT: The Standard for IoT Messaging.* Retrieved from <https://mqtt.org/>
* OReilly. (2020). *Motors.* Retrieved from <https://www.oreilly.com/library/view/beaglebone-cookbook/9781491915660/ch04.html>
* Bakker, B. d. (n.d.). *How to use a 16×2 character LCD with Arduino.* Retrieved from MarkersGuide.com: <https://www.makerguides.com/character-lcd-arduino-tutorial/>
* Campbell, S. (n.d.). *Basics of UART communication.* Retrieved from Circuit basics: <https://www.circuitbasics.com/basics-uart-communication/>
* tlemp.com. (2008, June). *MISRA C++:2008 - Guidelines for the use of the C++ language in critical systems.* Retrieved from <http://tlemp.com/download/rule/MISRA-CPP-2008-STANDARD.pdf>
* *Guidelines for the use of C language in vehicle based software.* (1998, April). Retrieved from <http://read.pudn.com/downloads327/doc/1441096/MISRA-C.pdf>
* Debaun, D. T. (2019, April 22). *Environmental Impact of Wireless Devices: How EMF Radiation is Endangering the Planet.* Retrieved from DefenderShield: <https://www.defendershield.com/environmental-impacts-emf-wireless-devices>
* Federal public service, health, food chain safety and environment. (2016, 12 1). *Ultrasound and high frequency sound.* Retrieved from <https://www.health.belgium.be/en/ultrasound-and-high-frequency-sound>
* Koilpillai, J. (2016, March 10). *Is MQTT Secure for the IoT? Only with an SDP!* Retrieved from Waverley Labs: <https://www.waverleylabs.com/is-mqtt-secure-for-the-iot-only-with-an-sdp/#:~:text=MQTT%20was%20designed%20for%20light,a%20two%2Dway%20hand%20shake>
* Kroes, P. (2005). *Ethical Issues in Engineering Design, Safety and Sustainability*. Anke van Gorp.Retrieved from <file:///C:/Users/User/Downloads/tpm_gorp_20051114.pdf>
* *Arduino Review.* (n.d.). Retrieved from CompareCamp: <https://comparecamp.com/arduino-ide-review-pricing-pros-cons-features/>
* Clarke, A. (n.d.). *BeagleBone Black Development Part 6 - Remote Application Debugging With Eclipse.* Retrieved from Embedded Coding: <http://blog.embeddedcoding.com/2015/05/beaglebone-black-development-part-6.html>
* *BeagleBone Black Eclipse and GDB. (n.d.).* Retrieved from Tincan tools: <https://www.tincantools.com/beaglebone-black-eclipse-gdb/>

**Instructor’s Remarks:**