**Cyclotron Vault Surveillance and Status Monitoring Robot**

**Variable Energy Cyclotron Centre, Kolkata**

**Submitted By**

**RISHAV SAHA**

**4th Year**

**Electronics & Communication Department**

**Institute of Engineering & Management**

**ACKNOWLEDGEMENT:**

I would like to express my heartiest gratitude towards my mentors and guides Mr. Tapas Kumar Bhaumik and Mr. Suman Guha of Variable Energy Cyclotron Centre, Kolkata without whose able guidance and advice this project would have ever been successful. I would also like to thank Dr. P. Y. Nabhiraj for providing me this wonderful opportunity to do my Summer Training in this esteemed institute of Variable Energy Cyclotron Centre, Kolkata.

……..……………………………..

**CERTIFICATION:**

**To whom ever it may concern,**

**This is to certify that Rishav Saha has worked under my guidance and the work represented in this project is done by him. I have read this project report and in my opinion it is fully adequate.**

**……………………………………**

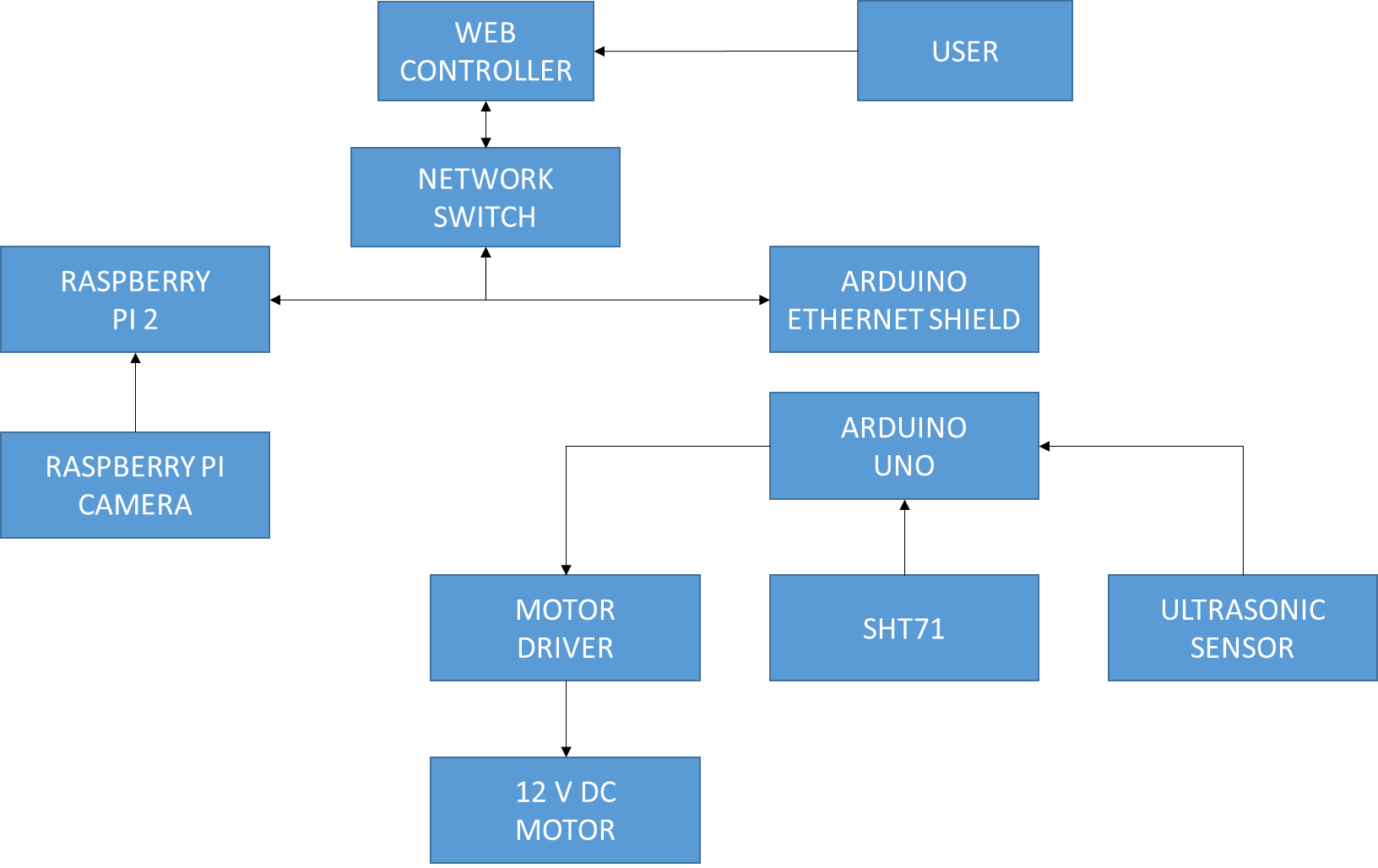
**INDEX:**

|  |  |  |
| --- | --- | --- |
| **SL No.** | **Topic** | **Page No.** |
| 1 | Introduction | 5 |
| 2 | Block Diagram of the Process | 6 |
| 3 | Detailed Hardware Description | 7-13 |
| 4 | Flowchart of the Software Program | 14 |
| 5 | Algorithm of the Software Program | 15 |
| 6 | Detailed Description of the Software Program | 15-18 |
| 7 | Screenshot of the Web Page and Live Video Streaming | 19 |
| 8 | Results | 20-22 |
| 9 | Conclusion | 23 |

**INTRODUCTION:**

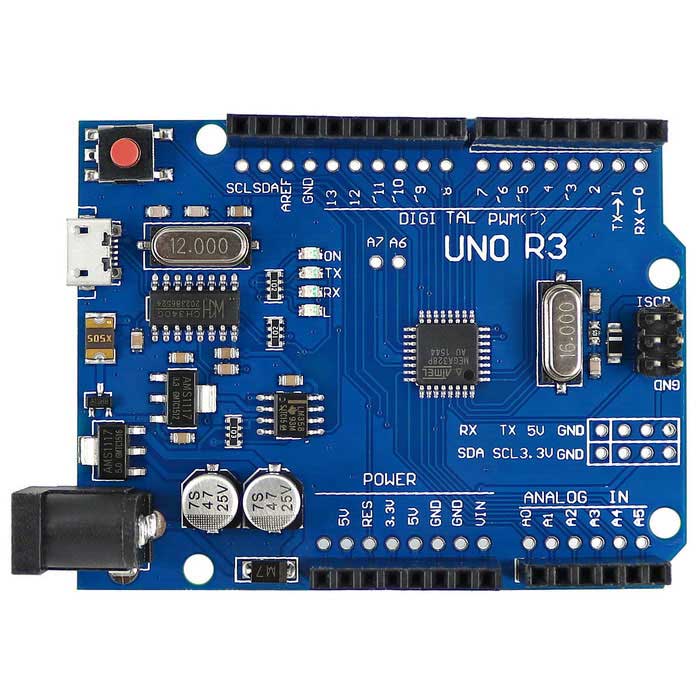
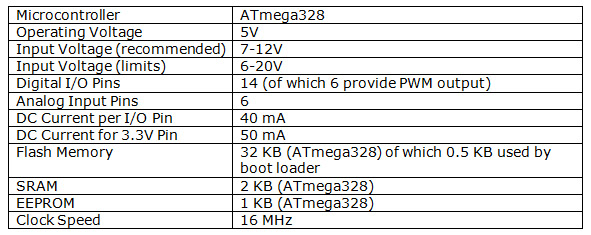
Room temperature cyclotron and Superconducting cyclotron of Variable Energy Cyclotron Centre Kolkata are the experimental facilities offered for the acceleration of multiply charged heavy positive ions as well as light ions at different energy levels. There exists high intensity of mixed radiation and neutron flux in the cyclotron vault, pit area during the operation of cyclotron. In order to reduce accidental radiation exposure the cyclotron vault, pit and ECR high bay area are inaccessible during the operation of the cyclotron. So, a mobile robot was designed and developed to acquire various field information including near real-time video from those inaccessible area with fixed obstacles. The robot is a two wheel dc motor driven robot which can be controlled via web server. The robot is mainly driven by an Arduino UNO board. Obstacle detecting ultrasonic sensors have been interfaced with the robot to measure the distance of obstacles from the robot. Also there is a Relative Humidity and Temperature Sensor onboard which takes notes of various field environmental conditions and updates a web page. Based on the values from the obstacle detector the controller can manually control the robot from the web page. In cases of emergency the robot has the ability to override the users command based on the Ultrasonic Sensors’ value. There is also a Raspberry Pi and a camera on the robot which streams the near Real Time video to the robot’s controller using a web based server application. The robot is unique as it can be used to monitor the inaccessible areas within the cyclotron vault while the cyclotron is operational without the risk of any accidental exposure to radiation.

**BLOCK DIAGRAM OF THE SYSTEM:**

****

**DETAILED DESCRIPTION OF HARDWARE:**

**Arduino Uno R3:** **Arduino Uno** is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller.

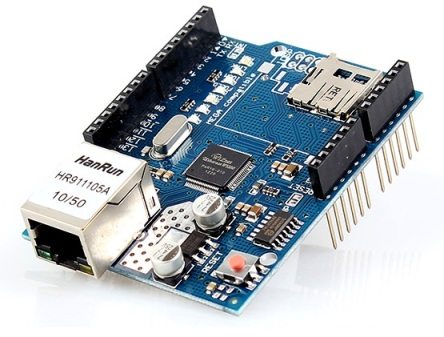
**Detailed Pin Map which has been used is given below:**

* **Pins A5, A4, A3 and A2** are used **for Driving the Motor.**
* **Pins 2 and** 3 are used for connecting the **Trigger** and **Echo Pin** of the **front Ultrasonic Sensor.**
* **Pins 5 and 6** are used for connecting the **Trigger and Echo Pin** of the **back Ultrasonic Sensor.**
* **Pin 11, 12 and 13** are used for the **SPI Connection** with the **Ethernet Shield.**
* **Pin 10** is used for **Chip Select** for the **Ethernet Shield.**
* **Pin 4** is used **for Chip Select** for the **SD Card Module.**
* **Pins 8 and 9** are used for the **Data** and **Clock Pins** of the **SHT71** module.

**Arduino Ethernet Shield 2:** The **Arduino Ethernet Shield 2** allows an Arduino Board to connect to the internet. It is based on the (Wiznet W5500 Ethernet chip). The Wiznet W5500 provides a network (IP) stack capable of robot TCP and UDP. It supports up to eight simultaneous socket connections. Use the Ethernet library to write sketches that connect to the Internet using the Shield. The Ethernet Shield 2 connects to an Arduino Board using long wire-wrap headers extending through the Shield. This keeps the pin layout intact and allows another Shield to be stacked on top of it.

The Ethernet Shield 2 has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard micro-SD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4.



**Ultrasonic Sensor (HC-SR04):** The **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are VCC, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple formula that

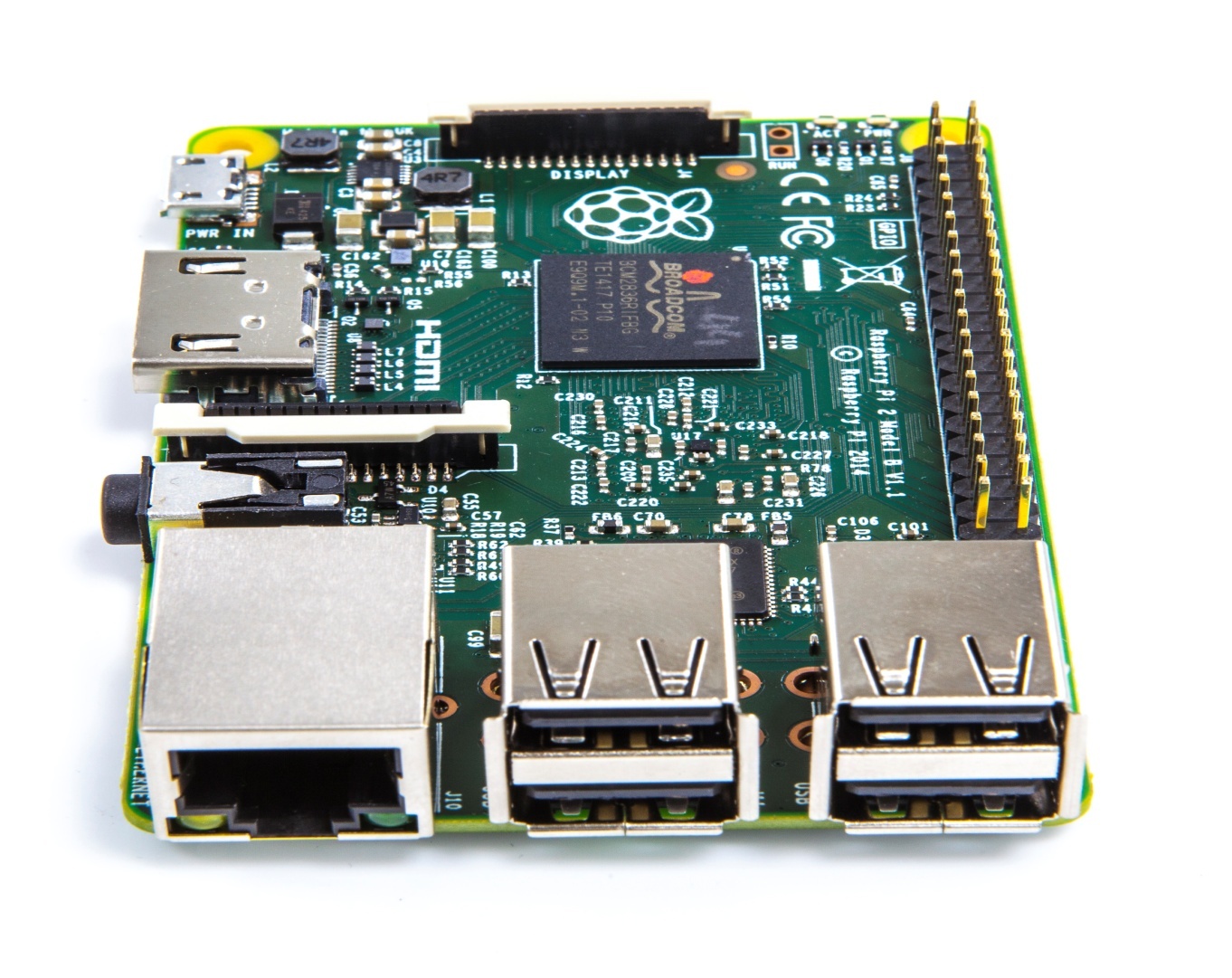
**Distance = Speed × Time**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.



**Raspberry Pi 2:** The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. The Raspberry Pi 2 board is a revision over the Raspberry Pi 1 board. The Raspberry Pi 2 board has the following specifications.

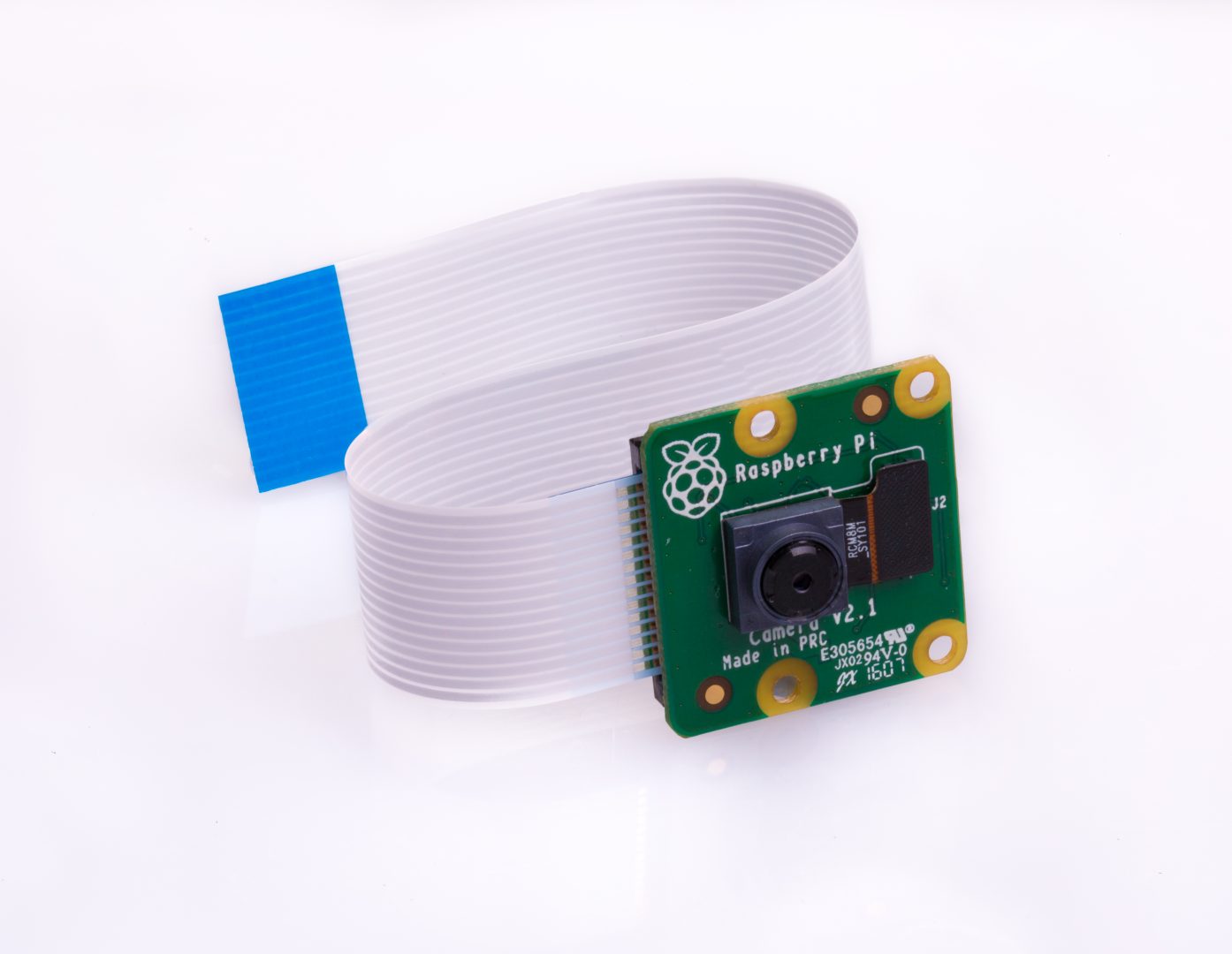
* SoC: Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
* CPU: 900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
* GPU: Broadcom Video Core IV @ 250 MHz
* More GPU info: OpenGL ES 2.0 (24 GFLOPS); 1080p30 MPEG-2 and VC-1 decoder (with license); 1080p30 h.264/MPEG-4 AVC high-profile decoder and encoder
* Memory: 1 GB (shared with GPU)
* USB ports: 4
* Video input: 15-pin MIPI camera interface (CSI) connector
* Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
* Audio input: I²S
* Audio outputs: Analog via 3.5 mm jack; digital via HDMI and I²S
* Storage: MicroSD
* Network: 10/100Mbps Ethernet
* Peripherals: 17 GPIO plus specific functions, and HAT ID bus
* Power rating: 800 mA (4.0 W)
* Power source: 5 V via MicroUSB or GPIO header



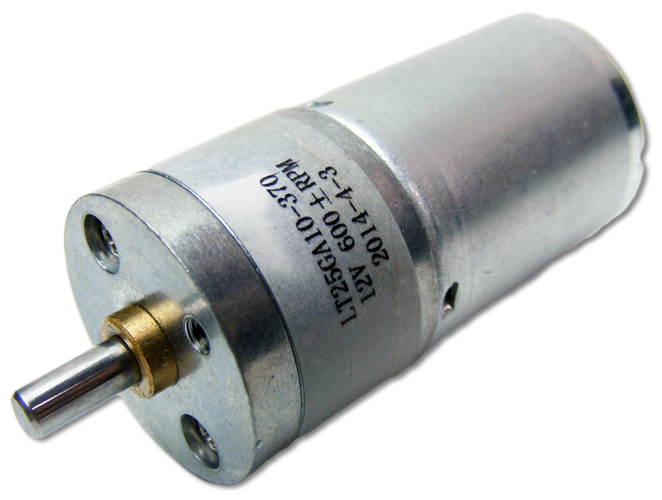
**Raspberry Pi Camera:** The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).

The Camera Module can be used to take high-definition video, as well as stills photographs.

 It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.



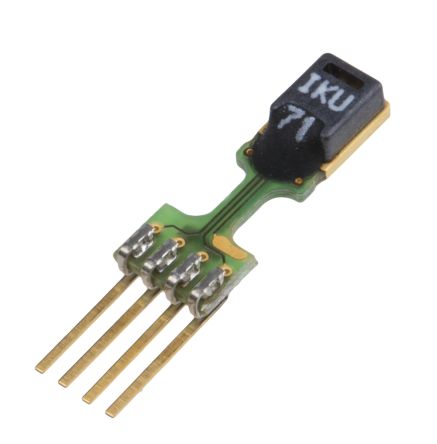
**DC Motors and Wheels:** Two 12V DC Motors have been used in this project, which have a speed of 10 rpm and high torque. These DC Motors are used to drive the robot and to change its direction using the differential drive mechanism.

**Motor Driver Circuit:** A motor driver circuit has been implemented using relays and transistors. The driver has the ability to run two motors simultaneously. The Motor Driver inputs and its output have been shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I/P-1 | I/P-2 | I/P-3 | I/P-4 | Robot Movement |
| 0 | 1 | 0 | 1 | FORWARD |
| 1 | 0 | 1 | 0 | BACKWARD |
| 1 | 0 | 0 | 1 | LEFT |
| 0 | 1 | 1 | 0 | RIGHT |

**SHT71:** It is a Sensirion’s relative humidity and temperature sensor with pins. The sensors integrate sensor elements plus signal processing in compact format and provide a fully calibrated digital output. A unique capacitive sensor element is used for measuring relative humidity while temperature is measured by a band-gap sensor. The applied CMOSens® technology guarantees excellent reliability and long term stability. The sensor is seamlessly coupled to a 14bit analog to digital converter and a serial interface circuit. This results in superior signal quality, a fast response time and insensitivity to external disturbances (EMC).



**Network Switch:** The Network Switch splits the incoming data from the Web Based Controller into the respective IP Addresses i.e. one for Arduino and another for the Raspberry Pi. It also acts as a common channel for robot the outgoing response from the Arduino and the Raspberry Pi Video Stream. The Network Switch used as of now is a wired Network Switch but it can be replaced with a Wireless one which will make allow us to control the robot wirelessly.



**FLOWCHART OF THE SOFTWARE PROGRAM:**

START

GET SENSOR DATA

NO

IS CLIENT CONNECTED?

YES

SEND WEB PAGE TO CLIENT

GOT REQUEST?

UPDATE

WEBPAGE

UPDATE

WEBPAGE

YES

GET DATA FROM ULTRASONIC SENSORS AND SHT71 SENSORS

CHECK DISTANCE FROM

ULTRASONIC SENSOR

SENSOR OR MOVEMENT REQUEST?

MOVEMENT

SENSOR

NO

IS DISTANCE >30cm?

YES

DO MOVEMENT

**ALGORITHM OF THE SOFTWARE PROGRAM:**

1. Start
2. Check if got request from client, if yes go to step 3 else go to step 2.
3. Send web page to client.
4. Check if got request from client, if yes go to step 5 else go to step 4.
5. Check if the request is for movement or sensor data. If movement go to step 6 else go to step 9.
6. Send the ultrasonic range, humidity and temperature data to client.
7. Check if the gap is below the threshold limit if yes go to step 4 else go to step 8.
8. Do movement and go back to step 4.
9. Send the ultrasonic range, humidity and temperature data to client and go back to step 4.

**Description of the Software Program:**

**Web Page:**

The Web page is designed with HTML and runs AJAX (Asynchronous JAVA and XML) Script in its background to refresh the sensor data value on the screen without actually refreshing the page. It also uses the same AJAX Technology to give instructions to the Robot about the movement.

The HTML Page has the following Components:

* **TITLE:** Shows ROBOT CONTROL
* **STATUS FIELD:** Shows the last movement done or attempted.
* **4 GAP FIELDS:** They show the data coming from the ultrasonic sensors. They change color to red if the data is below threshold else they show green color. Out of the 4 fields in the page only 2 are active to show the ultrasonic data as of now.
* **TEMPERATURE FIELD:** It shows the temperature value coming from the SHT71 Sensor.
* **RELATIVE HUMIDITY FIELD:** It shows the relative humidity value coming from the SHT71 Sensor.
* **4 MOVEMENT BUTTONS:** The 4 Movement Buttons FORWARD, BACKWARD, LEFT and RIGHT is used to send movement request to the robot using AJAX requests.

**Arduino Program:**

The Arduino Program implements the Algorithm discussed above. Some most important features of this program are.

* It uses the **HC-SR04 module** to measure distances before moving.
* It uses the **Ethernet Shield** to host a **Web Server** and to get the data sent from the computer which is received by the switch on the robot.
* It interacts with the **SHT71 sensor** using the I2C Protocol and sends the data via the Web Server.
* It uses a **SD Card Module** Embedded on the Ethernet Shield to store the **HTML Page** on a memory card.

1. **Some Important Libraries which were used:**

* **Ethernet:** For operating the Ethernet Shield.
* **SPI:** For SPI Communication.
* **SD:** For using the SD Card Module Functionalities.
* **Senserion:** For using the Temperature and Humidity Sensor.

1. **The functions defined in the program and their characteristics:**

* **void Setup() :** Contains all the initialization of the variables and pins.
* **Void Loop():** Contains the main loop over which the Arduino Program iterates endlessly.
* **Long getGap():** This function gets the data from the given ultrasonic Sensor and returns its value.
* **Void CheckDanger():**This function checks if the gap returned by the getGap() is less than the threshold value or not. If it is lesser it just sets a flag variable.
* **Void SendData():** This function is used to send data to the Web Page.
* **Void Forward():** Checks the forward flag value and moves the robot forward.
* **Void Backward():** Checks the backward flag value and moves the robot backward.
* **Void Left():** It rotates the robot anti clockwise.
* **Void Right():** It rotates the robot clockwise.
* **Void StrContains():**Is used to check if a HTTP Request contains a specific string.
* **Void StrClear():**Is used to clear a string.
* **void getHumid\_andTemp\_data():**Is used to get the values from SHT71 and send it to the Web Page.

Raspberry Pi Software:

# Raspbian Operating System: Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi.

1. **Tight VNC Server: Tight VNC** is afree VNC Software. **Virtual Network Computing** (**VNC**) is a graphical desktop sharing system that uses the Remote Frame Buffer protocol (RFB)to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network. It is used in this project to remotely access the Raspberry pi screen without the requirement of an external display.
2. **GStreamer:** GStreamer is a library for constructing graphs of media-handling components. The applications it supports range from simple Ogg/Vorbis playback, audio/video streaming to complex audio (mixing) and video (non-linear editing) processing.

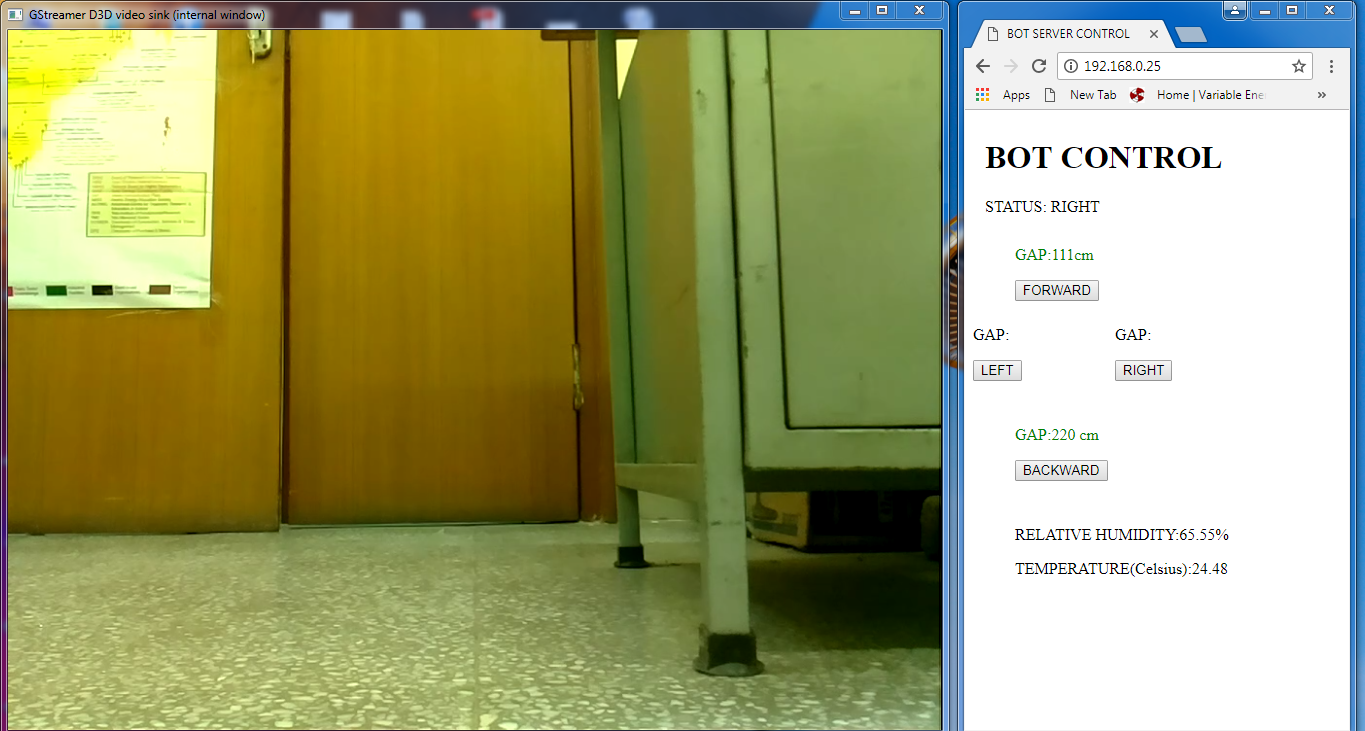
Applications can take advantage of advances in codec and filter technology transparently. Developers can add new codecs and filters by writing a simple plugin with a clean, generic interface.

The GStreamer is used with Raspberry Pi cam to stream the video from the Raspberry Pi camera.

**Special Features of the Software Programs :**

* AJAX Based Client Page which is much faster as compared to traditional refresh methods.
* Remote Monitoring System with real time Distance, Humidity and Temperature measurement.
* Real Time Video Streaming System with the use of GStreamer.
* Automatic Obstacle Detection and Over Riding of Master Commands to protect the robot from taking any unwanted move.
* Port Manipulation Based motor drive which gives much better control.
* Danger Indication in the web page.
* Status Indicator to track the last direction of movement.
* Real Time update of Humidity and Temperature values.

**Screenshot of the Web Page and Live Video Streaming:**



**RESULTS:**

**Forward Movement:**

|  |  |  |  |
| --- | --- | --- | --- |
| Observation No. | Distance Travelled | Time Taken | Mean Speed |
| 1 | 8cm | 2s | **4.5cm/s** |
| 2 | 9cm | 2s |
| 3 | 10cm | 2s |

**Backward Movement:**

|  |  |  |  |
| --- | --- | --- | --- |
| Observation No. | Distance Travelled | Time Taken | Mean Speed |
| 1 | 7cm | 2s | **4cm/s** |
| 2 | 8cm | 2s |
| 3 | 9cm | 2s |

**Right Movement:**

|  |  |  |  |
| --- | --- | --- | --- |
| Observation No. | Angle Rotated | Time Taken | Mean Angular Speed |
| 1 | 44° | 2s | **22°/s** |
| 2 | 43° | 2s |
| 3 | 45° | 2s |

**Left Movement:**

|  |  |  |  |
| --- | --- | --- | --- |
| Observation No. | Angle Rotated | Time Taken | Mean Angular Speed |
| 1 | 45° | 2s | **22°/s** |
| 2 | 43° | 2s |
| 3 | 44° | 2s |

**Ultrasonic Sensor Front:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Observation No. | Actual Distance | Measured Distance | Error | Mean Error |
| 1 | 14cm | 14cm | 0cm | **0cm** |
| 2 | 9cm | 9cm | 0cm |
| 3 | 5cm | 5cm | 0cm |

**Ultrasonic Sensor Back:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Observation No. | Actual Distance | Measured Distance | Error | Mean Error |
| 1 | 14cm | 14cm | 0cm | **0cm** |
| 2 | 13cm | 13cm | 0cm |
| 3 | 12cm | 12cm | 0cm |

**Temperature Sensor:**

|  |  |  |  |
| --- | --- | --- | --- |
| Actual Room Temperature | SHT71 Reading | Error | Error Percentage |
| 27.2°C | 27.97°C | 0.77°C | 2.83% |

**CONCLUSION:**

The Robot was successfully able to manoeuvre in the room upon receiving instruction from the user. It was also able to Stream Live Video. The ultrasonic sensors were able to detect obstacles and any movement command was overridden if the obstacle came within the threshold of the ultrasonic sensor. Relative Humidity, Temperature from the SHT71 sensor and the Distance from any obstacle as measured by the ultrasonic sensor could be updated in real time into the webpage with a refresh rate of 10sec and 30sec respectively.

**Present System’s Drawback:**

* Wired Network Switch: The Network Switch used is wired as of now. This makes free movement of the robot difficult and restricts the robot to a certain distance from the controller.
* The Arduino UNO development Board is not fast enough to acknowledge the web based requests.
* The Motors used have a low rpm which makes the robot slowly.
* There are no Ultrasonic Sensors on the sides of the robot.

**Future Scopes of Work:**

* The Wired Network Switch can be replaced by a wireless one which can improve the range and mobility of the robot.
* The Arduino UNO development Board can be replaced with a more powerful Arduino Mega development Board using the existing program which will improve upon the latency of the system.
* Faster and more powerful motors can be used to improve speed.
* Ultrasonic Sensors can be added on the sides of the robot.
* More sensors like Gas, Smoke, and Fire etc can be interfaced with the Arduino Mega Board.
* Better control mechanism can be implemented in the code to obtain finer level of control of the robot.
* The Raspberry Pi camera can be replaced with an Infrared one to detect any fires in the dark.