

**Smart Trash Bin Using Arduino Controller**

**Name: SULTAN SAMI YEHIA TUBAGI ID:438004392**

**Name: ABDULLAH ABDUALRHMAN ALMALKI ID: 437001432**

**Name: MOUHANAD IBRAHIM ALHAZMI ID: 434003241**

**Name: ABDULRHMAN ALI HAKAMI ID: 436007594**

**The supervision of Dr: Saleh Mohammed Baslama**

**Group: 4**

**DEPARTMENT OF (COMPUTER ENGINERING)**

**COLLEAG OF COMPUTER AND INFORMATION SYSTEMS**

**UM AL-QURA UNIVERSITY**

**UQU-CE-21-104**

**ABSTACT**

The rapid urbanization has raised the demand for more waste. This project aims at reducing this waste through the use of a smart bin. A sensor mounted on the top of the bin will determine the height of the object it's measuring. The device will also notify the user when it senses nearby objects. These smart bins will be used to manage waste effectively. The presence of foul smell from these wastes may lead to long term problems. It can also cause breeding of insects and diseases.

**Table of Content**

ABSTRACT……………………………………………………………………………….. 2

Table of Contents………………………………………………………………………………..3

LIST OF FIGURES……………………………………………………………………………..5

LIST OF Tables………………………………………………………………………………..6

LIST OF SYMBOLS/ABBREVIATIONS……………………………………………………..7

CHAPTER 1: Project Introduction………………………………………………………..1

Introduction………………………………………………………………………………..1

1.2 Project Objectives………………………………………………………………..1

1.3 Project Methodology………………………………………………………………2

1.4 Potential impact of project on society and Environment……………………………..2

CHAPTER 2: Literature overview and Project Characteristics…………………………..3

2.1 Introduction………………………………………………………………………………..3

2.2 literature overview……………………………………………………………………..3

2.3 Characteristics of the Project……………………………………………………………..4

Chapter 3: Project Design and Simulation……………………………………………………..5

3.3.1 Hardware design……………………………………………………………………..5

3.3.1.1 Control unit……………………………………………………………………………….5

3.3.1.2 Ultrasonic sensors………………………………………………………………………..7

3.3.1.3 Servo Motor…………………………………………………………………….11

3.3.2 Software design……………………………………………………………………14

3.3.2.1 Arduino IDE……………………………………………………………………………14

Chapter 4: Summary, Conclusion and Further Work …………………………………………..15

4.1 Summary………………………………………………………………………………..15

4.1 Conclusion………………………………………………………………………………..15

4.3 Future Work………………………………………………………………………………..15

Appendix………………………………………………………………………………..16

References………………………………………………………………………………..27

**List of Figures**

Figure 1 Arduino Uno 7

Figure 2 Ultrasonic Sensor HC SR04 8

Figure 3 Ultrasonic Sensor HC SR04 Pin Diagram 9

Figure 4 Ultrasonic Sensor - Working 10

Figure 5 2D model of the component 11

Figure 6 Servo Motor (MG996R) 12

Figure 7 (MG996R) Servo Motor Dimensions 12

Figure 8 Servo Motor Pinout (Wires) 12

**List of Tables**

[Table 1 Specifications of Arduino Uno 6](#_Toc68886744)

[Table 2 Ultrasonic Sensor Pin Configuration 8](#_Toc68886745)

[Table 3 Wire Configuration of Servo Motor (SG90) 13](#_Toc68886746)

**List of Symbols/Abbreviations**

RAM……………………………………………………………….……Random Access Memory ROM………………………………………...………………………………...Read Only Memory EAPROM……………………...…. Electronically Alterable Programmable Read-Only Memory

IR Sensor …………………………………………….……………………………. Infrared Sensor

IDE………………………………………………………………………Integrated Development

I/O pins …………………………………………………………………………Inputs & Outputs Pins

VDC………………………………………………………...…………….…Volts Direct Current

**CHAPTER 1: Project Introduction**

**Introduction**

Due to the increasing number of people living in a given area, the amount of garbage that ends up in the streets has also increased. In most cases, the cans of trash are regularly emptied.

Not enough time has been allocated for the proper disposal of the components. The level of the waste is determined by the sensors positioned in the area.

* 1. **Problem Statement**

Due to the increasing number of environmental issues in modern cities, the need for smart waste management systems has become prominent. Through the use of technology, these systems can help manage the waste disposal and reduce costs.

The goal of this work is to develop a smart trash bin that will allow users to optimize the process of trash collection. It will be useful for minimizing the cost of implementing such solutions.

The literature will introduce the reader to the various aspects of this work, including the main ideas and commercial solutions related to this project. The methodology and methods section also explains the various hardware and software utilized in this work.

* 1. **Project Objectives**

The goal of this project is to develop a prototype for an automatic open dustbin that automatically opens when someone wants to throw out their trash.

* 1. **Project Methodology**

This project methodology model takes into account the various phases of the project life cycle, such as conceptualization, design, development, validation, and evolution. It simplifies the work by separating the activities into separate phases.. Using a waterfall model as a project development methodology. Do to Specific system models, system architecture and detailed design of the project, to implementation process using Arduino IDE tool and aurdino UNO with C++ language for developing the modules in windows platform. In the smart dustbin hardware contains Aurdino UNO, Ultrasonic Sensor, Bread Board, ,Servo Motor ,Power Supply Connection Cables. In the smart dustbin Ultrasonic sensors will continuously monitor the status of the bin. This sensor will help to automatically sense nearby users and open the bin for them.

* 1. **Potential impact of project on society and Environment**

Being clean is important to maintain a clean and neat environment. This project aims to preserve the environment by keeping the dustbins clean and fresh.

**CHAPTER 2: Literature overview and Project Characteristics**

**2.1 Introduction**

Smart Bin is a garbage collector that's capable of monitoring the level of waste in its bin and sending alert messages to the local authorities.

This type of dustbins are very useful in areas where the frequency of using the dustbin changes. They can also be used with various features such as automatic closing and dimming.

An Arduino board is used to control the behavior. Power supply of 12V-2 Amps is used for the circuit. . Sensors are connected to the SPI Interface of the Arduino.

**2.2 Literature overview**

This project will introduce the concept of smart garbage bin, which will collect data using sensors and electronic modules. We reviewed the study which deal with the smart bin concepts. The review make up of different methods. In paper [1] discusses the different methodologies used to manage internet of things and describes the detailed functionality of IoT, and gives an overall idea of preparing application related to information management over internet. An overview of the concept for combining user application with IOT [2]. They outlined recent advancements by world foremost innovators in developing IoT Standards, big data management and mobile analytics, as well as standards and opensource platforms for developing IoT applications.

To realize the vision of the Internet of Things, we must address the various challenges that arise from the rapid emergence of new technologies and their impact on society.. All major successes in IoT research. A new technique was introduced in this paper and implement for smart city waste management connected with IOT [3], The dynamic scheduling concepts required for the cleaning of dustbin periodical and the Top-k query led us to priority-based cleaning of dustbins. City Garbage Collection Indicator using RF (Zigbee),, and GSM technology [4]. In the proposed system use a RFID, to identify the dustbin. It detects the dustbin fillings using Sensors and uses GSM to alert the authorities. Notable disadvantages are usage of RFID for identification and absence of log to record the data for each overflow. A Graphical user Interface GUI proposed in [5], which can show the current state of the dustbins. No efficient alert system is present since it sends all the alerts to one person.

**2.3 Characteristics of the Project**

Trash bins are small metal container that are use to store trash. They are used in streets, homes, parks, office etc. to collect the waste. In this project, We have designed Smart Dustbin using Arduino, Ultrasonic Sensor and Servo Motor, where the lid of the dustbin will automatically open itself upon detection of human hand.

The main concept behinds the Smart Dustbin use is Object Detection.

A similar methodology is implement here, where the Ultrasonic Sensor is place on top of the dustbin’s lid and when the sensor detect any objects like a human hand, it will force Arduino to open the lid.

**CHAPTER 3: Project Design and Simulation**

**3.3.1 Hardware design**

This project uses an Ultrasonic Sensor to create a robotic lidless dustbin. It uses the same methodology to avoid obstacles like human hand..

**3.3.1.1 Control unit**

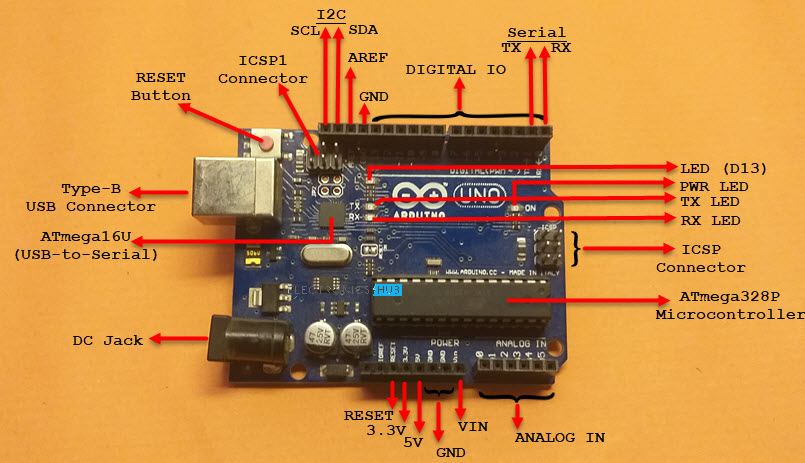
The ATmega328P is a 64-bit microcontroller board that has 14 digital input/output pins, 6 analog inputs, and a 16 MHz ceramic resonator. It's fully compatible with Windows.

The "Uno" board was the first in a series of USB-based Arduino boards. It was chosen to mark the release of the latest version of the software, which is called "Arduino 1.0".

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| LED\_BUILTIN | 13 |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

Table 1 Specifications of Arduino Uno

*Figure 1 Arduino Uno*



**3.3.1.2 Ultrasonic sensors**

This is HC-SR04 ultrasonic distance sensor. economical sensor provides 2 cm to 400 cm of non-contacts measurements functionality with a ranging accuracy that can reach up to 3mm.

There are 4 pin that you need to worry about on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground).

The sensor has control circuitry that can prevent inconsistent data depending on the application.

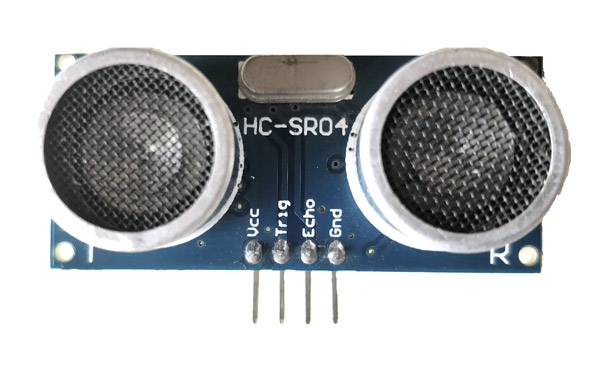


Figure 2 Ultrasonic Sensor HC SR04

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

Table 2 Ultrasonic Sensor Pin Configuration



Figure 3 Ultrasonic Sensor HC SR04 Pin Diagram

HC-SR04 Sensor Features:

* Operating voltage: +5V
* Theoretical Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

HC-SR04 Ultrasonic Sensor - Working

The HC-SR04 is a 4 pin module that has three pins labeled as Vcc, Echo, and Ground. It is commonly used for distance measuring and sensing.

Distance = Speed × Time

The transmitter sends an ultrasonic wave, when it gets objected to it gets reflected back to the sensor.



Figure 4 Ultrasonic Sensor - Working

To calculate the distance between two objects, first we need to know the speed and time of the US wave. Then, we need to know the time taken for the US wave to return.

This sensor is commonly used for monitoring distance in a wide range of devices such as mobile platforms, microcontrollers, and microprocessors. To start the measurement, it has to be made high for 10uS and then turned off. This will trigger the ultrasonic wave at frequency of 40Hz from the transmitter and the receiver must wait for the wave to return. Once the wave is returned the Echo pin go high for a amount of time which going to be equals to the time taken for the wave to return back to the sensor.

The sensor can be powered by its Vcc ad Ground pins. Its current consumption is less than 15mA and can be directly powered by the 5V pins of the microcontroller.

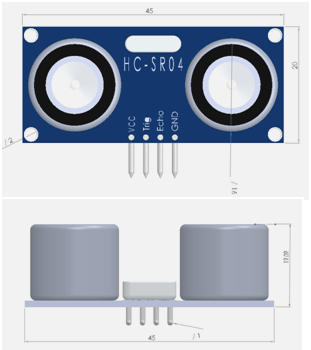
****

Figure 5 2D model of the component

**3.3.1.3 Servo Motor**

Due to their high torque, servo motors are commonly used in robots and other industrial applications. They can be electrically controlled to varying degrees of rotation.



Figure 6 Servo Motor (MG996R)

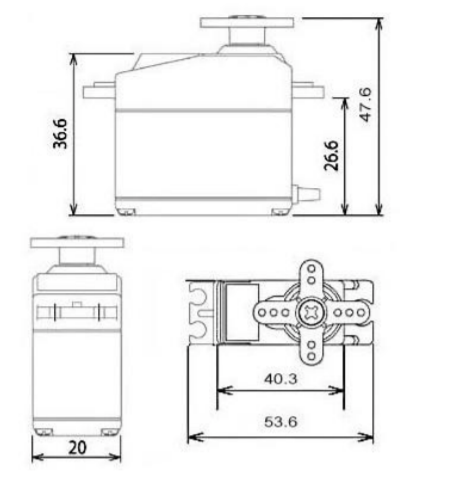
****

Figure 7 (MG996R) Servo Motor Dimensions

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

Figure 8 Servo Motor Pinout (Wires)

|  |  |  |
| --- | --- | --- |
| **Wire Number** | **Wire Colour** | **Description** |
| 1 | Brown | Ground wire connected to the ground of system |
| 2 | Red | Powers the motor typically +5V is used |
| 3 | Orange | PWM signal is given in through this wire to drive the motor |

Table 3 Wire Configuration of Servo Motor (MG996R)

Servo Motor (MG996R)Features

* Weight: 55 g
* Dimension: 40.7 x 19.7 x 42.9 mm approx.
* Stall torque: 9.4 kgf·cm (4.8 V ), 11 kgf·cm (6 V)
* Operating speed: 0.17 s/60º (4.8 V), 0.14 s/60º (6 V)
* Operating voltage: 4.8 V a 7.2 V
* Running Current 500 mA – 900 mA (6V)
* Stall Current 2.5 A (6V)
* Dead band width: 5 µs
* Stable and shock proof double ball bearing design
* Temperature range: 0 ºC –55 ºC

There are many types of servo motors available for different applications.

Most hobby servo motors are designed to operate from 4.8V to 6.5V, but they can also be operated at higher voltage levels. Generally, they are operated at 5V. If you're planning on using a hobby servo motor for a project, make sure that it can rotate at least from 0° to 360°, otherwise, it might not work for you.

The other important parameter that the motor has is its torque. Generally, the torque that the motor produces is measured in kilograms per cm. The most common choice is the 2.5 kg/cm torque which is provided by the MG996R.

**3.3.2 Software design**

C++ is a programming language that has many special features. Some of these include functions and methods. When creating a sketch, the name is used to represent the code.

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

**3.3.2.1 Arduino IDE**

The Arduino ecosystem is a community that thrives on the open-source platform. This is evidenced by the number of developers and organizations that support the platform.

The USB port of the Arduino board is used to connect it to a computer. The user then writes the code in the IDE and uploads it to the microcontroller, which can interact with various inputs and outputs.

There are a variety of free resources available to support both novice and experienced users. These include tutorials, guides, and materials. The open-source nature of the Arduino project makes it very easy to learn.

**CHAPTER 4: Summary, Conclusion Further Work**

**4.1 Summary**

The bin works by sensing the presence of a user and closing if the user leaves the operating range.

**4.2 Conclusion**

To minimize the accumulation of waste, various works have been carried out on the design and construction of smart bin systems. These systems will help us keep the bin clean and hygienic. They will also inform the authorities when the bin gets filled.

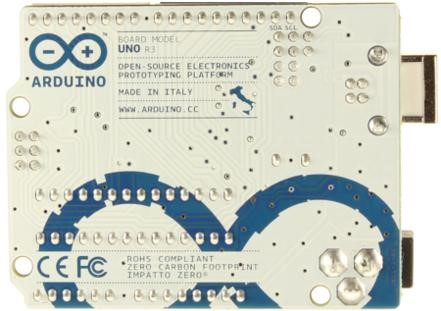
**4.3 Future work**

This prototype can be easily extended to other types of dustbins. It can be connected to any number of public dustbins and automatically collects waste when the lid is full.

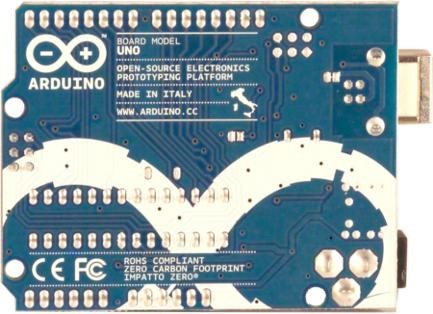
**Appendix**

**---Arduino Datasheet**

**---Arduino UNO**



*Arduino Uno R3 Front* *Arduino Uno R3 Back*



*Arduino Uno R2 Front* *Arduino Uno SMD* *Arduino Uno Front* *Arduino Uno Back*

**Overview**

The Arduino Uno is a 2-in-1 board that features 14 digital input/output pins, 6 analog inputs, and a 16 MHz ceramic resonator. It works seamlessly with the ATmega328 microcontroller.

The Uno differs from the others in that it doesn't use the FTDI USB-accelerated driver chip. Instead, it uses the Atmega16U2 USB-to-serial converter hardware.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode.](http://arduino.cc/en/Hacking/DFUProgramming8U2)

Revision 3 of the board has the following new features:

* This update also added two new pins that are near the IOREF pin and the SDA pin. These new pins allow the shields to adapt their voltage to the board.
* Stronger RESET circuit.
* Atmega 16U2 replace the 8U2.

**Summary**

|  |  |
| --- | --- |
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |

Input Voltage (recommended) 7-12V

|  |  |
| --- | --- |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock Speed | 16 MHz |

Schematic & reference design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](http://arduino.cc/en/uploads/Main/arduino_Uno_Rev3-02-TH.zip) (NOTE: works with Eagle 6.0 and newer) Schematic: [arduino-uno-Rev3-schematic.pdf](http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf)

**Note:** The Arduino reference design can use an Atmega8, 168, or 328, Current models use an

ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via an external power supply or with the USB connection. The power source is selected automatically.

AC-to-DC adapter or battery power can be used to provide external power to the board. The adapter can be connected to a wall-wart or battery.

The board can operate on a supply of up to 20 volts. If it is not supported by a voltage regulator, the board may become unstable.

. The power pins are as follows:

* **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
* **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND.** Ground pins.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library)](http://www.arduino.cc/en/Reference/EEPROM).

**Input and Output**

The 14 digital pins on the UNO can be used as an output or input. They operate at 5 volts and have a maximum output voltage of 40 A, some pins have specialized functions:

* **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
* **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt()](http://arduino.cc/en/Reference/AttachInterrupt) function for details.
* **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite()](http://arduino.cc/en/Reference/AnalogWrite) function.
* **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library.](http://arduino.cc/en/Reference/SPI)
* **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has six analog inputs, labeled A0, A5, A7, and A5. They provide 10 bits of resolution, some pins have specialized functionality:

* **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library.](http://arduino.cc/en/Reference/Wire)

There are a couple of other pins on the board:

* **AREF.** Reference voltage for the analog inputs. Used with [analogReference(](http://arduino.cc/en/Reference/AnalogReference)).
* **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports.](http://arduino.cc/en/Hacking/PinMapping168) The mapping for the Atmega8, 168, and 328 is identical.

**Communication**

The ATmega328 provides serial communication over USB, and an ATmega16U2 board can act as a virtual com port to a computer. The software for the board uses the standard USB drivers.

The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required.](http://arduino.cc/en/Guide/Windows#toc4) The Arduino software has a serial monitor that allows you to send and receive simple textual data from the board. The monitor will flash when the data is sent over a USB connection.

A [SoftwareSerial library](http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](http://arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library.](http://arduino.cc/en/Reference/SPI)

**Programming**

The Arduino Uno can be programmed with the Arduino software ([download)](http://arduino.cc/en/Main/Software). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](http://arduino.cc/en/Reference/HomePage) and [tutorials.](http://arduino.cc/en/Tutorial/HomePage)

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](http://arduino.cc/en/Tutorial/Bootloader) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference,](http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf) [C header files)](http://www.atmel.com/dyn/resources/prod_documents/avr061.zip).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](http://arduino.cc/en/Hacking/Programmer) for details.

The ATmega16U2 firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886) (Windows) or the [DFU programmer](http://dfu-programmer.sourceforge.net/) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1285962838) for more information.

**Automatic (Software) Reset**

Instead of having to physically press the reset button, the Arduino Uno can be reset by software running from a computer. This eliminates the need for an external button.

This line is used to reset the chip. The Arduino software use It is also allowing the user to upload code by simply clicking the upload button.

The Uno resets each time it is connected to a computer. It can also be reset by plugging it to a USB port. The bootloader is also programmed to ignore the first few bytes of data that are sent to it after a connection has been opened.

Make sure that the software that it uses waits a second before sending data to establish a connection.

The Uno contain a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1213719666/all) for details.

**USB Overcurrent Protection**

The Arduino Uno's resettable polyfuse is designed to protect USB ports from overcurrent and shorts. It will automatically break a connection if its current exceeds 500 mA.

**Physical Characteristics**

The maximum width and length of the Uno PCB are 2.1 and 2.7 inches respectively. Its USB connector and power jack can be extended beyond the former dimension.

**---Ultrasonic Datasheet**

**Ultrasonic Ranging Module HC - SR04**

**Product features:**

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

1. Using IO trigger for at least 10us high level signal,
2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
3. IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

**Wire connecting direct as following:**

5V Supply

Trigger Pulse Input

Echo Pulse Output

0V Ground

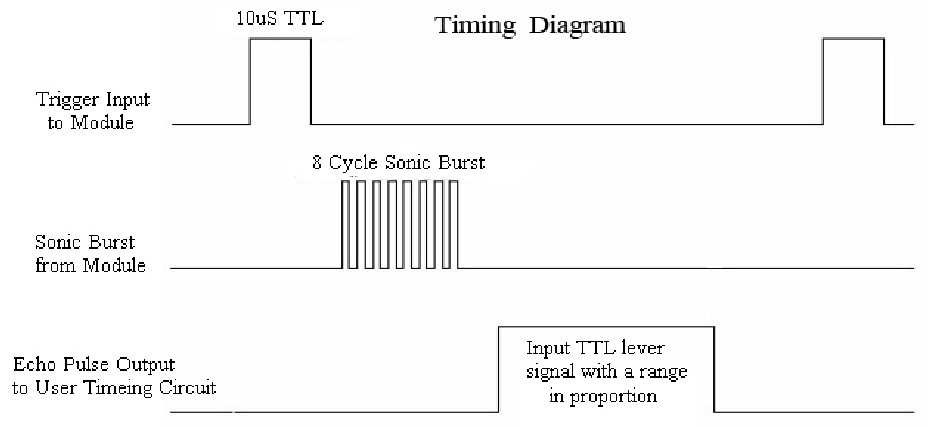
**Electric Parameter**

|  |  |
| --- | --- |
| **Working Voltage** | **DC 5 V** |
| **Working Current** | **15mA** |
| **Working Frequency** | **40Hz** |
| **Max Range** | **4m** |
| **Min Range** | **2cm** |
| **MeasuringAngle** | **15 degree** |
| **Trigger Input Signal** | **10uS TTL pulse** |
| **Echo Output Signal** | **Input TTL lever signal and the range in proportion** |
| **Dimension** | **45\*20\*15mm** |



**Timing diagram**

You only need to supply a 10uS pulse to trigger the range, and then the module will output an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The distance between the trigger and the echo signal is the width, and the range is the height. can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



**Attention**

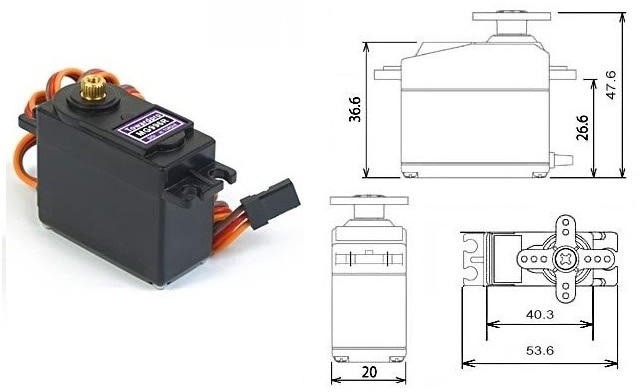
The module is not suggested to be connected to an electric supply, unless otherwise indicated.

When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise, it will affect the results of measuring.

**---Servo Motor Datasheet**

**MG996R High Torque**

**Metal Gear Dual Ball Bearing Servo**



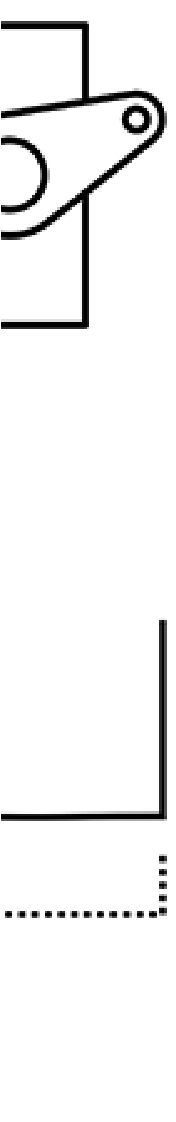
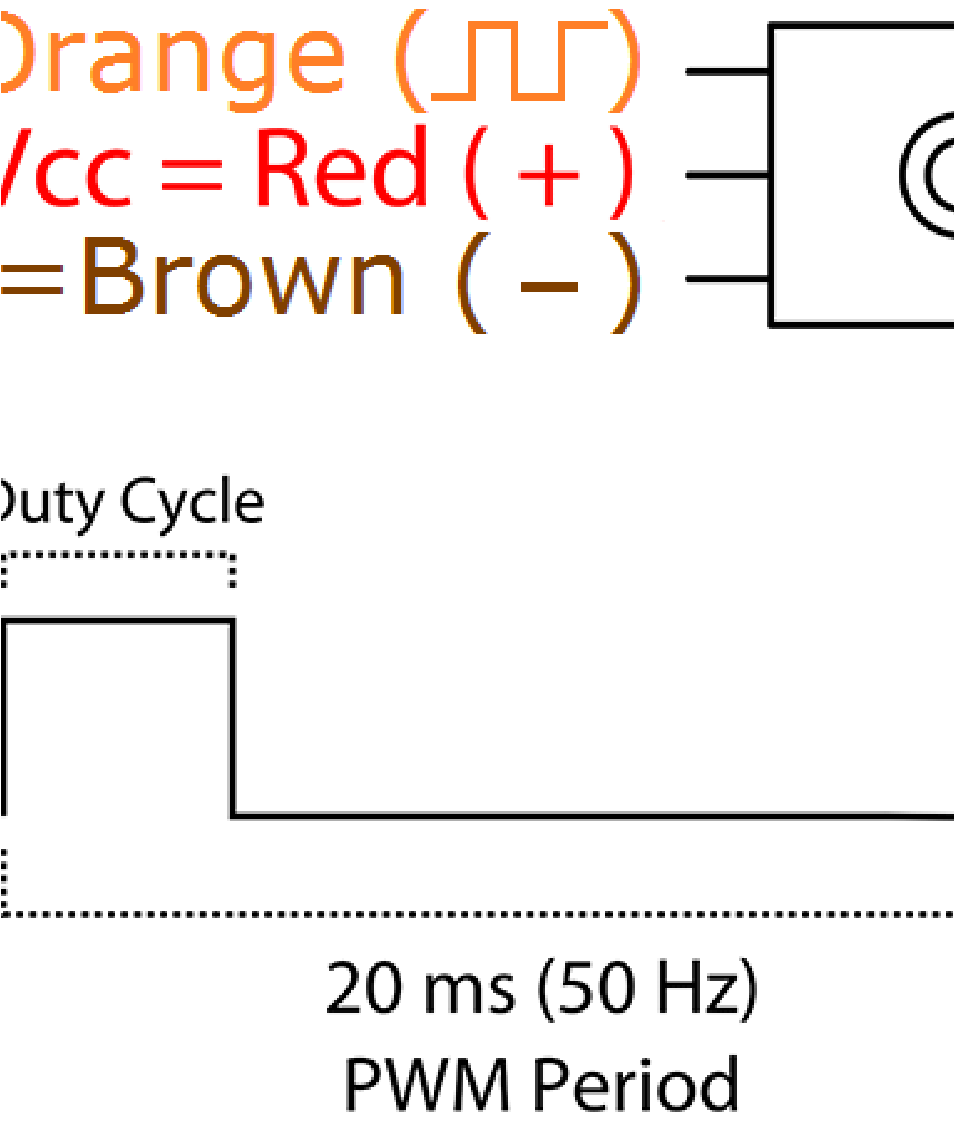
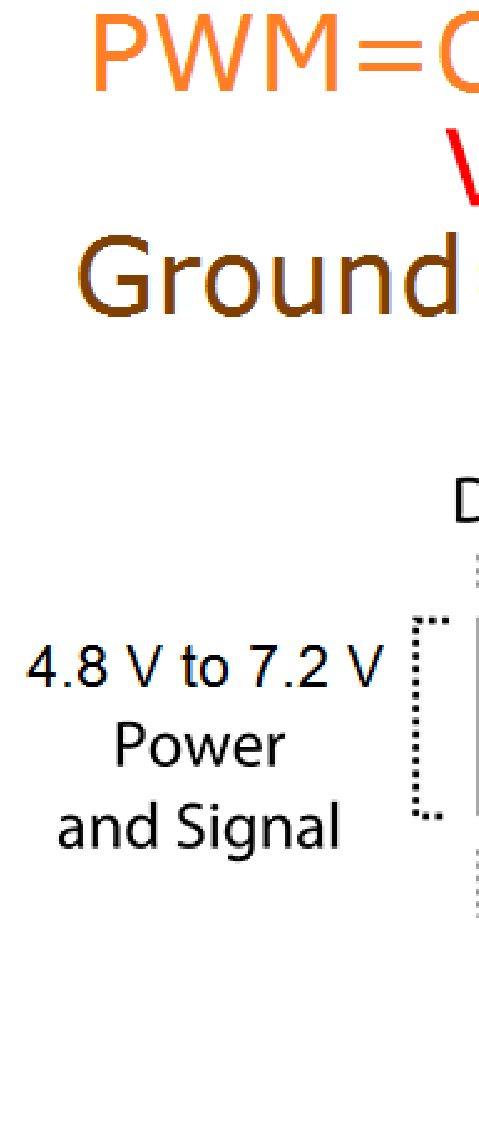
The MG996R features a high-torque metal gearing system that produces extra high 10 kg stalling torque. It's also more accurate than its predecessor.

This unit features upgraded gearing and motor for better dead bandwidth and centering. It also has 3 pin female header connectors for most receivers.

The MG996R Metal Gear Servo has a rotation speed of 120 degrees and can rotate at 120 degrees in each direction. It's a great starter servo for people who want to learn how to control it.

**Specifications**

* Weight: 55 g
* Dimension: 40.7 x 19.7 x 42.9 mm approx.
* Stall torque: 9.4 kgf·cm (4.8 V ), 11 kgf·cm (6 V)
* Operating speed: 0.17 s/60º (4.8 V), 0.14 s/60º (6 V)
* Operating voltage: 4.8 V a 7.2 V
* Running Current 500 mA – 900 mA (6V)
* Stall Current 2.5 A (6V)
* Dead band width: 5 µs
* Stable and shock proof double ball bearing design
* Temperature range: 0 ºC – 55 ºC



References

[1] “A state of the art reviews on The Internet of Things (IOT)”

P.Suresh1J. Vijay Daniel

2, R.H. Aswathy3 Dr.V.Parthasarathy4

International Conference on Science, Engineering and

Management Research (ICSEMR 2014).

[2] “Internet of Things: Challenges and State-of-the-art Solutions in

Internet-scale Sensor Information Management and Mobile

Analytics”, Arkady Zaslavsky, Dimitrios Georgakopoulos. 16th

IEEE International Conference on Mobile Data Management.

2015.

[3] “Top–k Query based Dynamic Scheduling for IOT enabled

Smart City Waste Collection” by Theodoros, Anagno

stopoulos1, Arkady. Zaslavsky 2, 1, Alexey Medvedev1, Sergei

Khoruzhnicov1. 16th IEEE International Conference on Mobile

Data Management. 2015.

[4] Vikrant Bhor, PankajMorajkar, Maheshwar Gurav and Dishant

Pandya. “Smart Garbage Management System”. International

Journal of Engineering Research & Technology (IJERT) ISSN:

2278-0181 IJERTV4IS031175 Vol. 4 Issue 03, March-2015.