International Journal of Research
Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 06 Issue 13
December 2019

IOT Agriculture System Based on LORAWAN

Sirigireddy Jhansi Rani[1], S.Mahaboob Basha[2]
[1]P.G. Scholar, [2]Guide, Head of the Department
[1,2] Branch:Embedded systems(ECE),M.Tech
Electronics and Communication Engineering Department
[1,2] Geethanjali College Of Engineering & Technology
Email: [1]jhansireddysirigireddy@gmail.com, [2]syedmahaboob45@gmail.com

Abstract

In the last years, besides the implementation in the smart city applications, IoT has also found significant place in the agricultural and food production process. In the paper we present an innovative, power efficient and highly scalable IoT agricultural system. This system is based on LORAWAN network for long range and low power consumption data transmission from the sensor nodes to the cloud services. Our system of cloud services is highly scalable and utilizes data stream for analytics purposes. In our case study we show some preliminary results for grape farm.

Keywords:- Temperature measurement, Security,Protocols,Agriculture, Production

Introduction

The recent advancement of the Internet of Things (IoT) in precision agriculture allows us to improve the overall agriculture management. The IoT is perfect match for precision agriculture due to its highly interoperable, scalable, pervasive and open nature. There are a lot of IoT derived technologies and all of them bring various benefits including reducing the risk of vendor lock-in, adopting machinery and better sensing/automation systems.

Every field, from health to environment, education to entertainment and industry to home is embracing the Internet of Things (IoT) revolution. Agriculture has seen many transformations and has adopted many machines to increase the yield. Field (soil and environmental parameters) and crop health monitoring are important factors for the yield to be of better quality and in larger quantities. In recent years, there has been many technological advancements in agriculture which have led to an increase in productivity and immunity of the crops. About 70% of the fresh water available in the world is consumed by the agriculture sector, with the help of soil moisture sensor we can optimize the irrigation process and use of water. The technology which plays a key role in this is the Internet of Things (IoT). Traditional agriculture is transforming into smart agriculture due to the penetration of the Internet of Things (IoT) in the agricultural sector. The IoT networks are reducing human labor requirements by monitoring crop health and field environment remotely. IoT uses a wireless sensor network (WSN) as the backbone for gathering information for these monitoring and control applications. The monitoring-system consists of end devices equipped with a variety of sensors to monitor various parameters like temperature, humidity, solar radiation, soil moisture, etc. and is capable of communicating this data to the other devices. IoT is helping the farmers by monitoring growth stages of the crop, diseases, and estimation of the yield by giving otherwise restricted low-power, low-cost devices access to greater processing capabilities via the Internet.

Being motivated by the above benefits and potentials of IoT applied in precision agriculture, considering the non-existence of complete reliable well-established and standard solution yet, we designed our model. In this paper we will propose a solution that provides easy and inexpensive network scalability, while maintaining the possibility of placing sensor nodes at a great distance from the base station which reduces the complexity and the cost of the overall network.

Our model architecture is highly customizable and provides data analytics solution, enables large-scale data processing on real time observation streams of data coming from variety of sources, such as sensory network, weather forecasting services, etc. In our solution LoRaWAN LPWAN (Low-Power Wide-Area Network) as a transmission protocol that is designed by the LoRa Alliance and meets the need of the IoT services.

It has easy implementation, out of the box security layer and ensures maximum coverage of hundreds of square meters. It requires minimal maintenance with low power consumption which makes it ideal for large number of sensors. All of above makes LoRaWAN perfect for the Internet of Things (IoT) in various fields and especially in agriculture, where the extension of the fields and their distribution across vast areas require an ad hoc protocol.

LITERATURE SURVEY

Industrial automation using CAN protocol describe project is implemented to control the industrial loads that are run by DC motor based on the temperature variations of the process. Various process control systems are depends on the temperature. So this project achieves this with the use of CAN protocol which is highly efficient and reliable low-cost communication. Two microcontrollers are used in this project, one for acquiring temperature data and the other for controlling the DC motor. CAN Controller MCP2515 and CAN transceiver MCP2551 are connected to both microcontrollers to implement CAN communication for exchanging the data but disadvantage practically it is limited to 110 nodes due to the hardware transceivers. It supports cabling up to 250 meters.

Industrial automation using ZigBee describe the transmitter section, the Zigbee module is configured in such a way that it receives the data collected from the microcontroller and sends it to the remote receiver. In this system, the microcontroller is programmed to collect the data from an analog to digital converter that continuously monitors temperature, voltage and current parameters. At the receiver side, the Zigbee module receives all the sent data from a Zigbee transmitter within the range of communication. This data is further transferred to the microcontroller using an embedded circuitry wherein the microcontroller program compares all these data parameters with predefined set limits. If any parameter exceeds its limit, then the microcontroller sends command signals to a relay driver IC, which is responsible to operate different loads such as motors, relays, circuit breakers, etc. All these parameters’ information is also displayed on LCD display as a Human machine interface. In this way, industrial parameters can be easily monitored and controlled through the short range low cost and low powered Zig bee communication technology.

It supports two ways communication between transmitting devices and controllers at 10-100 meters distance.Monitoring and Irrigation Automation system. In their work, to monitor crop-field a system is developed by using sensors and according to the decision from a server based on sensed data, the irrigation system automated. By using wireless transmission the sensed data forwarded towards to web server database. If irrigation is automated then that means if the moisture and temperature fields fall below of the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to the user.

Second part is hardware interface module, which provides appropriate interface to sensors and actuator of home automation system. Unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface modules as long as it exists on WiFi network coverage. System supports a wide range of home automation devices like power management components, and security components. The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

HARDWARE IMPLEMENTATION OF THE PROJECT

This chapter briefly explains about the Hardware Implementation of the project. It discusses the design and working of the design with the help of block diagram and circuit diagram and explanation of circuit diagram in detail. It explains the features, timer programming, serial communication, interrupts of atmega328 microcontroller. It also explains the various modules used in this project.

4.1 Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC’s to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The project design and principle are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of the design and working condition is explained using circuit diagram and system wiring diagram.

INTRODUCTION TO MICROCONTROLLER

Based on the Processor side Embedded Systems is mainly divided into 3 types

1. Micro Processor : - are for general purpose eg: our personal computer

2. Micro Controller:- are for specific applications, because of cheaper cost we will go for these

3. DSP (Digital Signal Processor):- are for high and sensitive application purpose

MICROCONTROLLER VERSUS MICROPROCESSOR

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

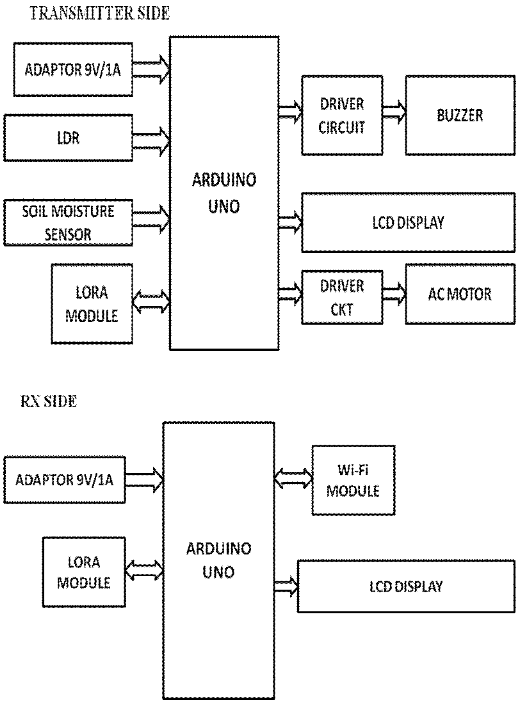
Microprocessor vs. Microcontroller	
Microprocessor	**Microcontroller**
CPU is stand alone RAM, RAM, I/O, timer are separate	CPU,RAM,ROM,I/O and timer are all on a single chip
Designer can decide on the amount of ROM, RAM and I/O ports.	Fix amount of on chip ROM,RAM, I/O Ports.
Expansive, Versatility	For applications in which cost, power and space are critical
General purpose	Single purpose

Table: 4.1 Microprocessor versus Microcontroller

4.1.1 Block Diagram of the Project

The block diagram of the design is as shown in Fig. The brief description about block diagram is given below.

Block diagram of proposed system



Power Supply

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

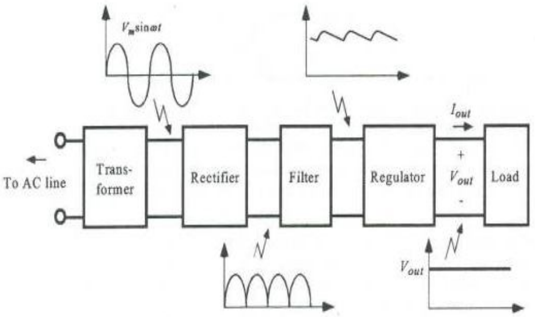


Fig: Power supply

Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

Voltage regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

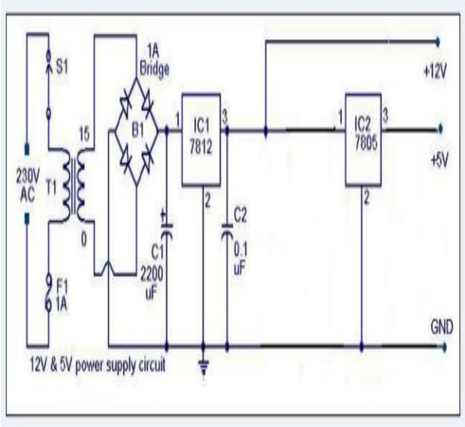


Fig: Power supply circuit diagram

ARDUINO UNO

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino. Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.

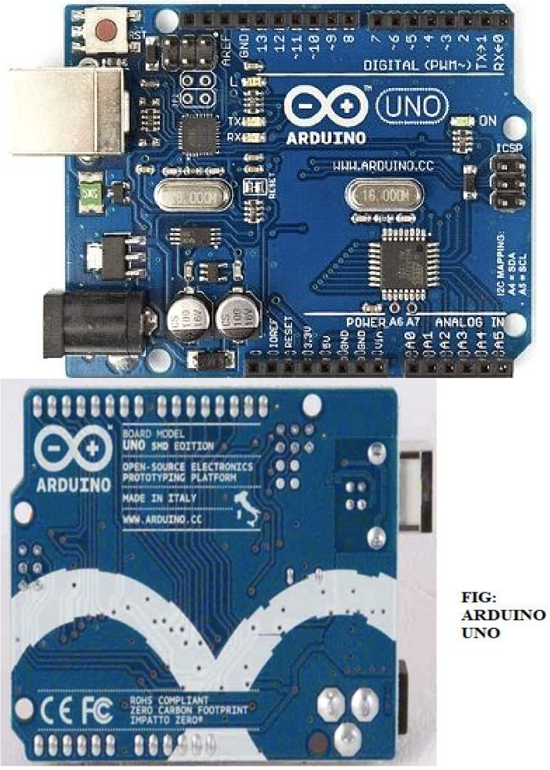


FIG: ARDUINO UNO

ARDUINO UNO-R3 PHYSICAL COMPONENTS

ATMEGA328P-PU microcontroller

The most important element in Arduino Uno R3 is ATMEGA328P-PU is an 8-bit Microcontroller with flash memory reach to 32k bytes.

Features

1. High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family
   - Advanced RISC Architecture

– 131 Powerful Instructions
– Most Single Clock Cycle Execution
– 32 x 8 General Purpose Working Registers
– Fully Static Operation
– Up to 20 MIPS Throughput at 20MHz
– On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
– 32KBytes of In-System Self-Programmable Flash program

2. Memory

– 1KBytes EEPROM
– 2KBytes Internal SRAM
– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
– Data Retention: 20 years at 85°C/100 years at 25°C(1)
– Optional Boot Code Section with Independent Lock Bits
• In-System Programming by On-chip Boot Program
• True Read-While-Write Operation
– Programming Lock for Software Security
• Atmel® QTouch® Library Support
– Capacitive Touch Buttons, Sliders and Wheels
– QTouch and QMatrix® Acquisition
– Up to 64 sense channels
3. Atmel-42735B-ATmega328/P_Datasheet_Complete-11/2016
• Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
– Real Time Counter with Separate Oscillator
– Six PWM Channels
– 8-channel 10-bit ADC in TQFP and QFN/MLF package
• Temperature Measurement
– 6-channel 10-bit ADC in PDIP Package
• Temperature Measurement
– Two Master/Slave SPI Serial Interface
– One Programmable Serial USART
– One Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
– Programmable Watchdog Timer with Separate On-chip Oscillator
– One On-chip Analog Comparator
– Interrupt and Wake-up on Pin Change
• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated Oscillator
– External and Internal Interrupt Sources
– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and
4. Extended Standby
• I/O and Packages
– 23 Programmable I/O Lines
– 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
• Operating Voltage:
– 1.8 - 5.5V
• Temperature Range:
– -40°C to 105°C
• Speed Grade:
– 0 - 4MHz @ 1.8 - 5.5V
– 0 - 10MHz @ 2.7 - 5.5V
– 0 - 20MHz @ 4.5 - 5.5V
• Power Consumption at 1MHz, 1.8V, 25°C
– Active Mode: 0.2mA
– Power-down Mode: 0.1µA
– Power-save Mode: 0.75µA (Including 32kHz RTC)

OTHER ARDUINO UNO R3 PARTS

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms.

In addition, 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.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13

(SCK). These pins support SPI communication using the SPI library.

- **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 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e.1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library. There are a couple of other pins on the board:
- **AREF:** Reference voltage for the analog inputs. Used with 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.

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).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library 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 for details. For SPI communication, use the SPI library.

Description

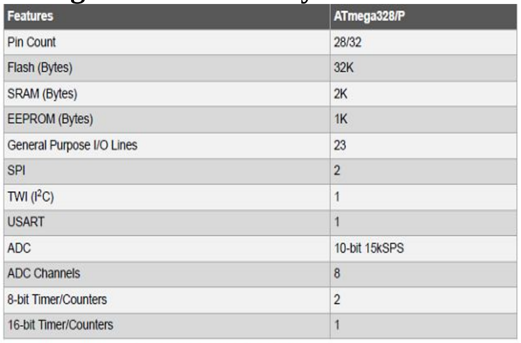
The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel’s high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core.

The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications. The ATmega328/P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

Configuration Summary



Features	ATmega328/P
Pin Count	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
General Purpose I/O Lines	23
SPI	2
TWI (I^{2}C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

Table: Arduino Configurations

Block Diagram

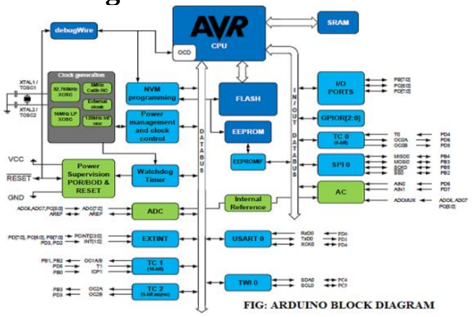


FIG: ARDUINO BLOCK DIAGRAM

Pin Configurations

Pin-out

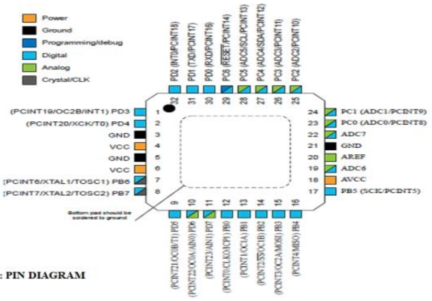


FIG: PIN DIAGRAM

Pin Descriptions

VCC - Digital supply voltage

GND – Ground

Port B (PB[7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB[7:6] is used as TOSC[2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC[5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC[5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is un programmed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset. The various special features of Port C are elaborated in the *Alternate Functions of Port C* section.

Port D (PD[7:0])

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

AVCC is the supply voltage pin for the A/D Converter, PC[3:0], and PE[3:2]. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC[6:4] use digital supply voltage, VCC.

AREF

AREF is the analog reference pin for the A/D Converter.

ADC[7:6] (TQFP and VFQFN Package Only)

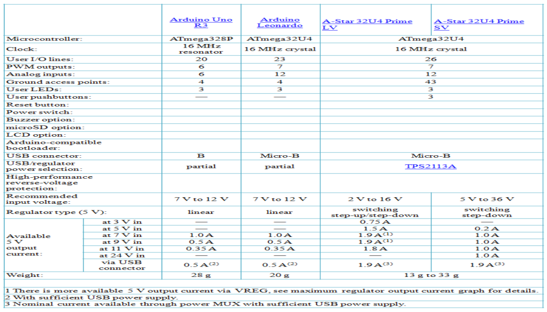
In the TQFP and VFQFN package, ADC[7:6] serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

I/O Multiplexing

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively it can be assigned to one of the peripheral functions. The following table describes the peripheral signals multiplexed to the PORT I/O pins.

Choosing the right controller

The table below compares the Arduino Uno, Leonardo, and our A-Star 32U4 Prime controllers. The A-Star Primes are based on the same ATmega32U4 AVR microcontroller as the Leonardo and ship with Arduino-compatible bootloaders. The Primes also offer many advantages, including superior power management that enables efficient operation from 2.7 V to 11.8 V (LV version) or 5 V to 36 V (SV version).

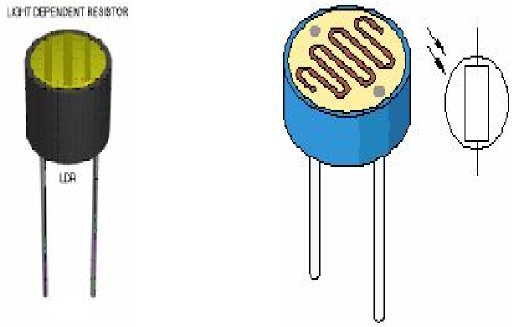


		Arduino Uno R3	Arduino Leonardo	A-Star 32U4 Prime LV	A-Star 32U4 Prime SV
Microcontroller:		ATmega328P	ATmega32U4	ATmega32U4	
Clock:		16 MHz resonator	16 MHz crystal	16 MHz crystal	
User I/O lines:		20	23	26	
PWM outputs:		6	7	7	
Analog inputs:		6	12	12	
Ground access points:		4	4	43	
User LEDs:		3	3	3	
User pushbuttons:		—	—	3	
Reset button:					
Power switch:					
Buzzer option:					
microSD option:					
LCD option:					
Arduino-compatible bootloader:					
USB connector:		B	Micro-B	Micro-B	
USB/regulator power selection:		partial	partial	TPS2113A	
High-performance reverse-voltage protection:					
Recommended input voltage:		7 V to 12 V	7 V to 12 V	2 V to 16 V	5 V to 36 V
Regulator type (5 V):		linear	linear	switching step-up/step-down	switching step-down
Available 5 V output current:	at 3 V in	—	—	0.75 A	—
	at 5 V in	—	—	1.5 A	0.2 A
	at 7 V in	1.0 A	1.0 A	1.9 A(1)	1.0 A
	at 9 V in	0.5 A	0.5 A	1.9 A(1)	1.0 A
	at 11 V in	0.35 A	0.35 A	1.8 A	1.0 A
	at 24 V in	—	—	—	1.0 A
	via USB connector	0.5 A(2)	0.5 A(2)	1.9 A(3)	1.9 A(3)
Weight:		28 g	20 g	13 g to 33 g	

1 There is more available 5 V output current via VREG, see maximum regulator output current graph for details.
2 With sufficient USB power supply.
3 Nominal current available through power MUX with sufficient USB power supply.

Light Dependent Resistor:

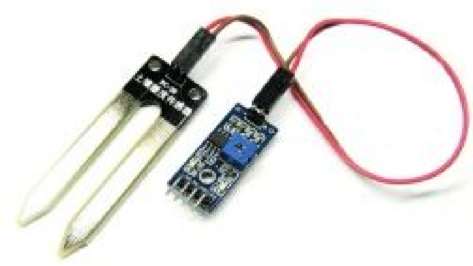
LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1,000,000 ohms, but when they are illuminated with light, the resistance drops dramatically. Thus in this project, LDR plays an important role in switching on the lights based on the intensity of light i.e., if the intensity of light is more (during daytime) the lights will be in off condition. And if the intensity of light is less (during nights), the lights will be switched on.



The output of the LDR is given to ADC which converts the analog intensity value into corresponding digital data and presents this data as the input to the microcontroller.

Moisture sensor:

The **Moisture sensor** is used to measure the water content (moisture) of soil. When the soil is having water shortage, the module output is at high level, else the output is at low level. This sensor reminds the user to water their plants and also monitors the moisture content of soil. It has been widely used in agriculture, land irrigation and botanical gardening.



Specifications:

- Working Voltage:**5V**
- Working Current:**<20mA**
- Interface type: **Analog**
- Working Temperature:**10°C~30°C**

1.1.1. Working of Sensor

The soil moisture sensor consists of two probes which are used to measure the volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value.

When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.

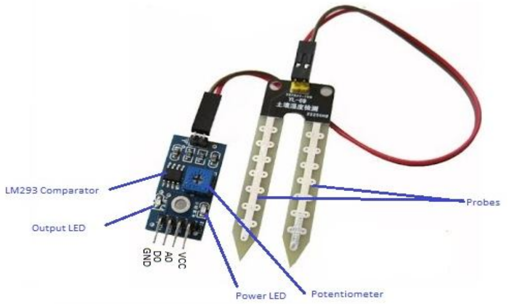
This sensor can be connected in two modes; Analog mode and digital mode.

Pin Out – Soil Moisture Sensor

The soil Moisture sensor FC-28 has four pins

- VCC: For power
- A0: Analog output
- D0: Digital output
- GND: Ground

The Module also contains a potentiometer which will set the threshold value and then this threshold value will be compared by the LM393 comparator. The output LED will light up and down according to this threshold value.



1.1.2. Analog Mode – Interfacing Soil Moisture Sensor with controller

To connect the sensor in the analog mode, we will need to use the analog output of the sensor. When taking the analog output from the soil moisture sensor FC-28, the sensor gives us the value from 0-1023. You can further set different ranges of the moisture values and turn on or off the water pump according to it. This analog output is connected to ADC pin of the microcontroller or external ADC channel input if the microcontroller does not contain inbuilt ADC.

1.1.3.

1.1.4. Digital Mode – Interfacing Soil Moisture Sensor with the microcontroller

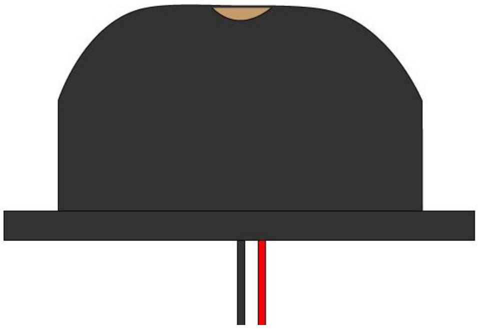
To connect the soil moisture sensor FC-28 in the digital mode, we will connect the digital output of the sensor to the digital pin of the microcontroller. The Sensor module contains a potentiometer with it, which is used to set the threshold value. This threshold value is then compared with the sensor output value using the LM393 comparator which is placed on the sensor module.

The LM393 comparator will compare the sensor output value and the threshold value and then gives us the output through the digital pin. When the sensor value will be greater than the threshold value, then the digital pin will give us 5V and the LED on the sensor will light up and when the sensor value will be less than this threshold value, then the digital pin will give us 0V and the light will go down.

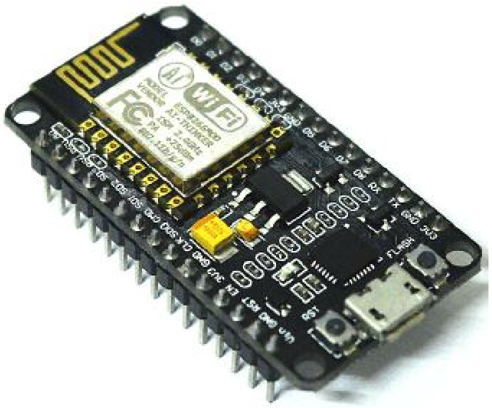
Buzzer:
The piezo buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits.

The buzzer produces a same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other. This, push and pull action, results in a sound wave. Most buzzers produce sound in the range of 2 to 4 kHz.

The Red lead is connected to the Input and the Black lead is connected to Ground.



NodeMCU ESP8266 ESP-12E WiFi Development Board
NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term “NodeMCU” by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.



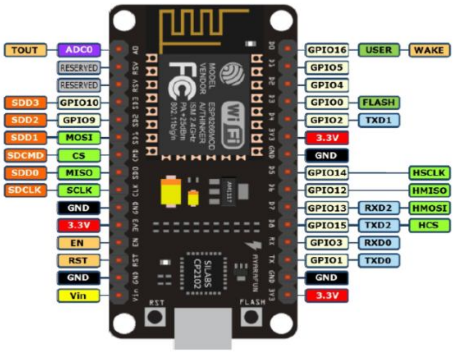
Features
Version: DevKit v1.0
Breadboard Friendly
Light Weight and small size.
3.3V operated, can be USB powered.
Uses wireless protocol 802.11b/g/n.
Built-in wireless connectivity capabilities.
Built-in PCB antenna on the ESP-12E chip.
Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.

Uses CP2102 USB Serial Communication interface module.

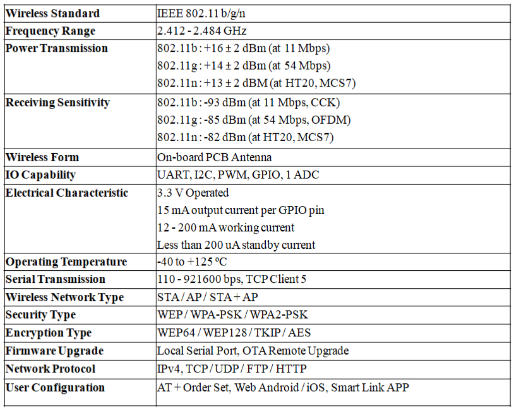
Arduino IDE compatible (extension board manager required).

Supports Lua (alike node.js) and Arduino C programming language.

Pinout:



Specifications of ESP-12E WiFi Module



Wireless Standard	IEEE 802.11 b/g/n
Frequency Range	2.412 - 2.484 GHz
Power Transmission	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBM (at HT20, MCS7)
Receiving Sensitivity	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
Wireless Form	On-board PCB Antenna
IO Capability	UART, I2C, PWM, GPIO, 1 ADC
Electrical Characteristic	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
Operating Temperature	-40 to +125 ºC
Serial Transmission	110 - 921600 bps, TCP Client 5
Wireless Network Type	STA / AP / STA + AP
Security Type	WEP / WPA-PSK / WPA2-PSK
Encryption Type	WEP64 / WEP128 / TKIP / AES
Firmware Upgrade	Local Serial Port, OTA Remote Upgrade
Network Protocol	IPv4, TCP / UDP / FTP / HTTP
User Configuration	AT + Order Set, Web Android / iOS, Smart Link APP

LIQUID CRYSTAL DISPLAY:

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

1. The declining prices of LCDs.
2. The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.
4. Ease of programming for characters and graphics.

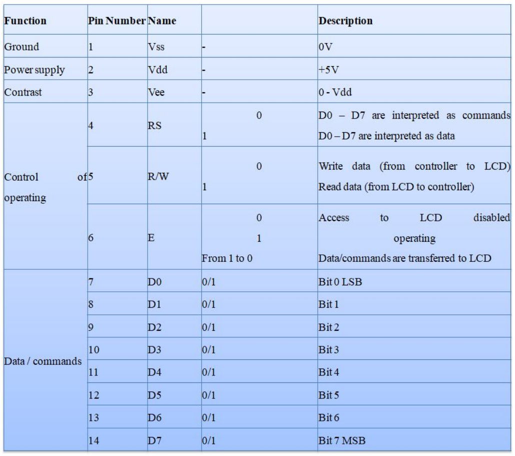
These components are "specialized" for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

A model described here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 microcontroller (*Hitachi*) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic shifting message on display

(shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

Pins Functions

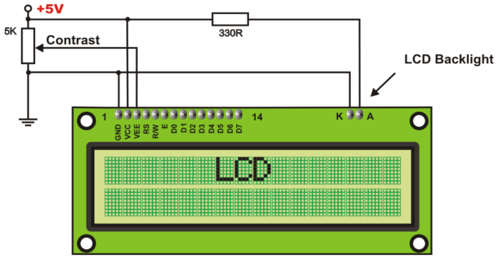
There are pins along one side of the small printed board used for connection to the microcontroller. There are total of 14 pins marked with numbers (16 in case the background light is built in). Their function is described in the table below:



Function	Pin Number	Name		Description
Ground	1	Vss	-	0V
Power supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0 - Vdd
Control of operating	4	RS	0 1	D0 – D7 are interpreted as commands D0 – D7 are interpreted as data
	5	R/W	0 1	Write data (from controller to LCD) Read data (from LCD to controller)
	6	E	0 1 From 1 to 0	Access to LCD disabled operating Data/commands are transferred to LCD
Data / commands	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB

LCD screen:

LCD screen consists of two lines with 16 characters each. Each character consists of 5x7 dot matrix. Contrast on display depends on the power supply voltage and whether messages are displayed in one or two lines. For that reason, variable voltage 0-Vdd is applied on pin marked as Vee. Trimmer potentiometer is usually used for that purpose. Some versions of displays have built in backlight (blue or green diodes). When used during operating, a resistor for current limitation should be used (like with any LE diode).

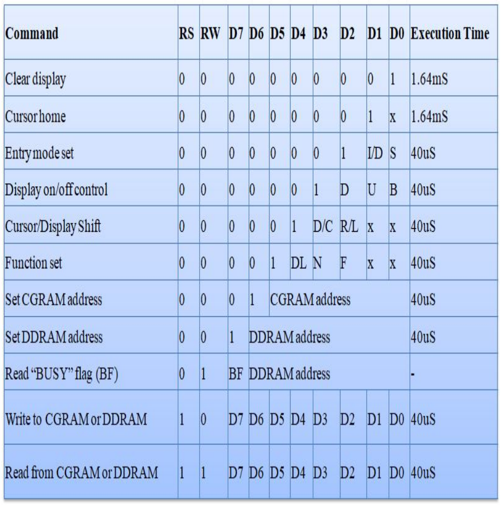


LCD Basic Commands

All data transferred to LCD through outputs D0-D7 will be interpreted as commands or as data, which depends on logic state on pin RS:

RS = 1 - Bits D0 - D7 are addresses of characters that should be displayed. Built in processor addresses built in "map of characters" and displays corresponding symbols. Displaying position is determined by DDRAM address. This address is either previously defined or the address of previously transferred character is automatically incremented.

RS = 0 - Bits D0 - D7 are commands which determine display mode. List of commands which LCD recognizes are given in the table below:



Command	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Execution Time
Clear display	0	0	0	0	0	0	0	0	0	1	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40uS
Display on/off control	0	0	0	0	0	0	1	D	U	B	40uS
Cursor/Display Shift	0	0	0	0	0	1	D/C	R/L	x	x	40uS
Function set	0	0	0	0	1	DL	N	F	x	x	40uS
Set CGRAM address	0	0	0	1	CGRAM address						40uS
Set DDRAM address	0	0	1	DDRAM address							40uS
Read "BUSY" flag (BF)	0	1	BF	DDRAM address							-
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40uS
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40uS

I/D 1 = Increment (by 1)
R/L 1 = Shift right
0 = Decrement (by 1)
0 = Shift left
S 1 = Display shift on
DL 1 = 8-bit interface
0 = Display shift off
0 = 4-bit interface
D 1 = Display on
N 1 = Display in two lines
0 = Display off
0 = Display in one line
U 1 = Cursor on
F 1 = Character format 5x10 dots
0 = Cursor off
0 = Character format 5x7 dots
B 1 = Cursor blink on
D/C 1 = Display shift
0 = Cursor blink off
0 = Cursor shift

LCD Initialization:

Once the power supply is turned on, LCD is automatically cleared. This process lasts for approximately 15mS. After that, display is ready to operate. The mode of operating is set by default. This means that:

1. Display is cleared

2. Mode

DL = 1 Communication through 8-bit interface

N = 0 Messages are displayed in one line

F = 0 Character font 5 x 8 dots

3. Display/Cursor on/off

D = 0 Display off

U = 0 Cursor off

B = 0 Cursor blink off

4. Character entry

ID = 1 Addresses on display are automatically incremented by 1

S = 0 Display shift off

Automatic reset is mainly performed without any problems. Mainly but not always! If for any reason power supply voltage does not reach full value in the course of 10mS, display will start perform completely unpredictably. If voltage supply unit can not meet this condition or if it is needed to provide completely safe operating, the process of initialization by which a new reset enabling display to operate normally must be applied.

Algorithm according to the initialization is being performed depends on whether connection to the microcontroller is through 4- or 8-bit interface. All left over to be done after that is to give basic commands and of course- to display messages.

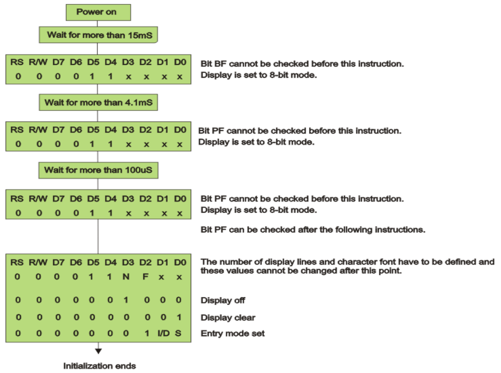
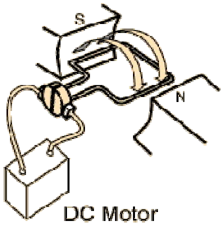
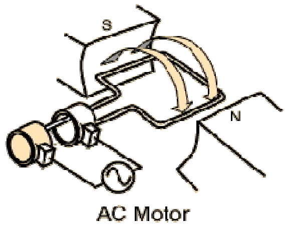

Power on
Wait for more than 15mS
RS R/W D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0 1 1 x x x x
Bit BF cannot be checked before this instruction.
Display is set to 8-bit mode.
Wait for more than 4.1mS
RS R/W D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0 1 1 x x x x
Bit PF cannot be checked before this instruction.
Display is set to 8-bit mode.
Wait for more than 100uS
RS R/W D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0 1 1 x x x x
Bit PF cannot be checked before this instruction.
Display is set to 8-bit mode.
Bit PF can be checked after the following instructions.
RS R/W D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0 1 1 N F x x
The number of display lines and character font have to be defined and these values cannot be changed after this point.
0 0 0 0 0 0 1 0 0 0
Display off
0 0 0 0 0 0 0 0 0 1
Display clear
0 0 0 0 0 0 0 1 I/D S
Entry mode set
Initialization ends

Fig: Procedure on 8-bit initialization.

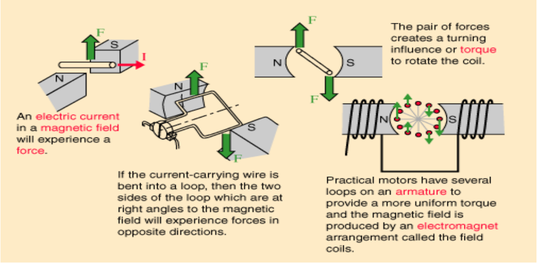
MOTORS

Electric motors involve rotating coils of wire which are driven by the magnetic force exerted by a magnetic field on an electric current. They transform electrical energy into mechanical energy.





Working of electric motors


F
S
N
I
F
N
S
F
An electric current in a magnetic field will experience a force.
If the current-carrying wire is bent into a loop, then the two sides of the loop which are at right angles to the magnetic field will experience forces in opposite directions.
F
N
S
F
The pair of forces creates a turning influence or torque to rotate the coil.
N
S
Practical motors have several loops on an armature to provide a more uniform torque and the magnetic field is produced by an electromagnet arrangement called the field coils.

Electric motors are used to efficiently convert electrical energy into mechanical energy. Magnetism is the basis of their principles of operation. They use permanent magnets, electromagnets, and exploit the magnetic properties of materials in order to create these amazing machines. There are several types of electric motors available today. The following outline gives an overview of several popular ones. There are two main classes of motors: AC and DC. AC motors require an alternating current or voltage source (like the power coming out of the wall outlets in your house) to make them work. DC motors require a direct current or voltage source (like the voltage coming out of batteries) to make them work. Universal motors can work on either type of power. Not only is the construction of the motors different, but the means used to control the speed and torque created by each of these motors also varies, although the principles of power conversion are common to both.

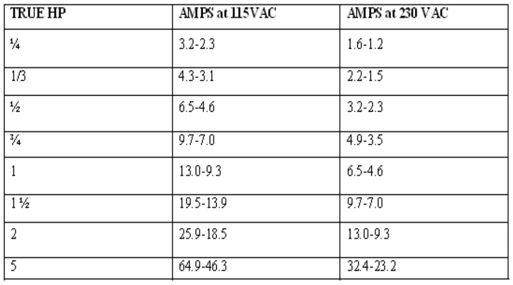
In industry, motors are used to move, lift, rotate, accelerate, brake, lower and spin material in order to coat, paint, punch, plate, make or form steel, film, paper, tissue, aluminum, plastic and other raw materials. They range in power ratings from less than 1/100 hp to over 100,000 hp. The rotate as slowly as 0.001 rpm to over 100,000 rpm. They range in physical size from as small as the head of a pin to the size of a locomotive engine.

Horsepower & RPM

Horsepower

Electric motors are rated by horsepower, the home shop will probably utilize motors from 1/4 HP for small tools and up to 5 HP on air compressors. Not all motors are rated the same, some are rated under load, others as peak horsepower and hence we have 5 HP compressors with huge motors and 5 Hp shopvacs with tiny little motors. Unfortunately all 5 HP compressor motors are not equal in actual power either, to judge the true horsepower the easiest way is to look at the amperage of the motor. Electric motors are not efficient, most have a rating of about 50% due to factors such as heat and friction and some may be as high as 70%.

This chart will give a basic idea of the true horse power rating compared to the ampere rating. Motors with a higher efficiency rating will draw fewer amps, for example a 5 HP motor with a 50% efficiency rating will draw about 32 amps at 230 VAC compared to about 23 amps for a motor with a 70% rating.



TRUE HP	AMPS at 115VAC	AMPS at 230 VAC
¼	3.2-2.3	1.6-1.2
1/3	4.3-3.1	2.2-1.5
½	6.5-4.6	3.2-2.3
¾	9.7-7.0	4.9-3.5
1	13.0-9.3	6.5-4.6
1 ½	19.5-13.9	9.7-7.0
2	25.9-18.5	13.0-9.3
5	64.9-46.3	32.4-23.2

A quick general calculation when looking at a motor is 1 HP = 10 amps on 110 volts and 1 HP = 5 amps on 220 volts.

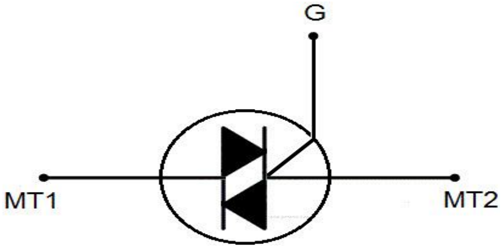
RPM

The shaft on a typical shop motor will rotate at either 1725 or 3450 RPM (revolutions per minute). The speed of the driven machine will be determined by the size of pulleys used, for example a 3450 RPM motor can be replaced by a 1750 RPM motor if the diameter of the pulley on the motor is doubled. The opposite is true as well but if the pulley on the 1750 RPM motor is small it is not always possible to replace it with one half the size. It may be possible to double the pulley size on the driven machine if it uses a standard type of pulley, (not easily done on air compressors for example).

Electronic speed reducers such as the ones sold for routers will not work on induction type motors.

TRIAC

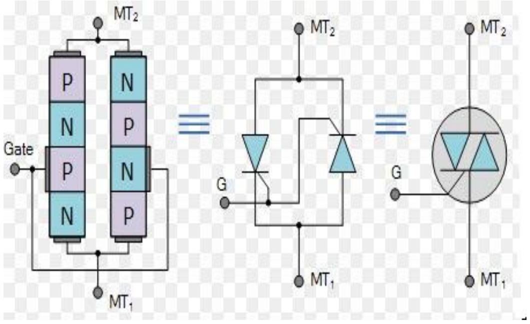
The TRIAC is an ideal device used for the AC switching applications. This can control the flow of current over both halves of an alternating cycle. The Thyristor can control only over the one-half of a cycle. The other remaining half no conduction occurs and accordingly only half the waveform can be utilized.



1.1.5. Construction of TRIAC

The TRIAC consist of four layers like PNPN is in the positive direction and the negative direction consists of NPNP as we can see in the figure. The three-terminal bidirectional device blocks the current in the

OFF state and it will act as an open circuit switch.



The four triggering modes of operations can operate by using the I-V characteristics of TRIAC.

Quadrant I operation: VMT2, positive; VG1 positive

Quadrant II operation: VMT21 positive; VGl negative

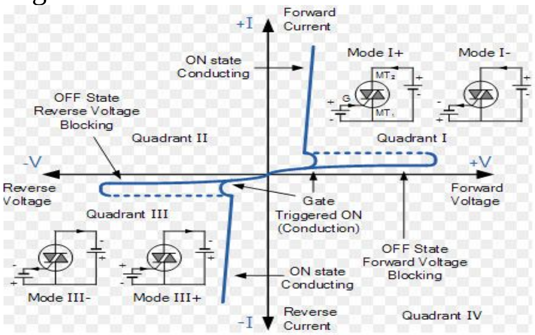
Quadrant III operation: VMT21 negative; VGl negative

Quadrant IV operation: VMT21 negative; VG1 positive

where VMT21 and VGl are the voltages of the terminal of MT2 and gate with respect to terminal MT1.

I-V Characteristics

The following figure shows the typical TRIAC characteristics. The Triode For Alternating Current consists of ON and OFF state characteristics which are similar to the SCR. Now the characteristics are applicable to both the voltages of positive and negative.



I-V Characteristics of TRIAC

Advantages of TRIAC

- In the TRIACs, there will be single gate control conduction in both the directions.
- If the voltage is decreased to zero the TRIAC turns OFF
- Basically, it is bidirectional device and in the both the directions it will conduct.
- High voltage rating are available

Disadvantages

- It doesn't fire symmetrically on both sides of the waveforms
- It doesn't appropriate for the DC power
- It has very high switching delay.

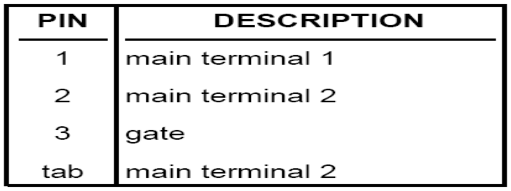
Applications

The TRIACs are using many electrical switching applications like

- The electric fan speed control
- Smaller motor controls
- Domestic light dimmer
- Control of small AC powered domestic appliances

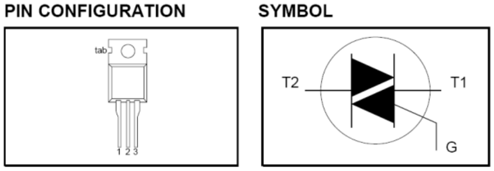
BT136 Description

Glass passivated triacs in a plastic envelope, intended for use in applications requiring high bidirectional transient and blocking voltage capability and high thermal cycling performance. Typical applications include motor control, industrial and domestic lighting, heating and static switching.



PIN	DESCRIPTION
1	main terminal 1
2	main terminal 2
3	gate
tab	main terminal 2

Pin description



PIN CONFIGURATION

SYMBOL

LORA COMMUNICATION:

LoRa is a '**Lo**ng **Ra**nge' low power wireless standard intended for providing a cellular style low data rate communications network.

Aimed at the M2M and IoT market, LoRa is ideal for providing intermittent low data rate connectivity over significant distances. The radio interface has been designed to enable extremely low signal levels to be received, and as a result even low power transmissions can be received at significant ranges.

The LoRa modulation and radio interface has been designed and optimised to provide exactly the type of communications needed for remote IoT and M2M nodes.

LoRa technology basics

There are several key elements of LoRa technology. Some of its key features include the following:

- Long range: 15 - 20 km.
- Millions of nodes
- Long battery life: in excess of ten years

There are various elements to LoRa technology that provide the overall functionality and connectivity for the system:

LoRa PHY / RF interface: The LoRa physical layer or PHY is key to the operation of the system. It governs the aspects of the RF signal that is transmitted between the nodes or end points, i.e. the sensors and the LoRa gateway where signals are received. The physical layer or radio interface governs aspects of the signal including the frequencies, modulation format, power levels, signalling between the transmitting and receiving elements,

LoRa protocol stack: In addition to the LoRa physical layer, the LoRa Alliance has also defined an open protocol stack. The creation of the open source stack has enabled the concept of LoRa to grow because all the different types of companies involved in LoRa development, use and deployment have been able to come together to create an easy to use, low cost solution for connectivity all manner of connected IoT devices.

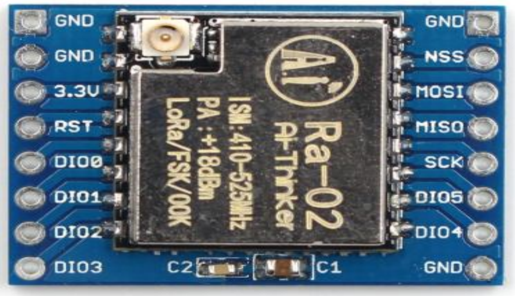
LoRa network architecture (LoRaWAN): Apart from the RF elements of the LoRa wireless system, there are other elements of the network architecture, including the overall system architecture, backhaul, server and the application computers. The overall architecture is often referred to as LoRaWAN.

***1.2.* Typical applications**

LoRa wireless technology is ideally placed to be used in a wide variety of applications. The low power and long range capabilities mean that end points can be deployed in a wide variety of places, in buildings and outside and still have the capability of being able to communicate with the gateway. As such the system is easy to deploy and it can be used for a large number of Internet of Things, IoT and machine to machine, M2M, applications.

Applications for LoRa wireless technology include: smart metering; inventory tracking, vending machine data and monitoring; automotive industry; utility applications infact anywhere where data reporting and control may be needed. LoRa technology is particularly attractive for many applications because of its long range capability. New nodes can easily be connected and activated and coverage is easy to provide.

LORA-02 MODULE:



Features:

- LoRa ™ spread spectrum modulation technology
- Receive sensitivity as low as -141 dBm
- Excellent resistance to blocking
- Supports preamble detection
- Supports half-duplex SPI communication
- Programmable bit rate up to 300Kbps

- Supports FSK, GFSK, MSK, GMSK, LoRa ™ and OOK modulation modes
- Supports automatic RF signal detection, CAD mode and ultra high speed AFC
- Packets with CRC, up to 256 bytes
- Small package with double volume stamps

Description:

Ra-02 can be used for ultra-long distance spread spectrum communication, and compatible FSK remote modulation and demodulation quickly, to solve the traditional wireless design cannot take into account the distance, anti-interference and power consumption. Ra-02 can be widely used in a variety of networking occasions, for automatic meter reading, home building automation, security systems, remote irrigation systems, is the ideal solution for things networking applications. Ra-02 is available in SMD package and can be used for rapid production by standard SMT equipment. It provides customers with high reliability connection mode.

Product Specifications:

Module Model	Ra-02
Package	SMD-16
Size	17*16*(3.2 ± 0.1) mm
Interface	SPI
Programmable bit rate	UP to 300Kbps
Frequency Range	410-525 MHz
Antenna	IPEX
Max Transmit Power	18±1 dBm
Power (Typical Values)	433MHz: TX:93mA RX:12.15mA Standby:1.6Ma 470MHZ: TX:97mA RX:12.15mA Standby:1.5mA
Power Supply	2.5~3.7V'Typical 3.3V
Operating Temperature	-30 ć ~ 85 ć
Storage Environment	-40 ć ~ 90 ć , < 90%RH
Weight	0.45g

FIRMWARE IMPLEMENTATION OF THE PROJECT DESIGN

FIRMWARE IMPLEMENTATION

This chapter briefly explains about the firmware implementation of the project. The required software tools are discussed in below section.

Software Tool Required

Arduino 1.0.6 software tools used to program microcontroller. The working of software tool is explained below in detail.

PROGRAMMING MICROCONTROLLER

A compiler for a high level language helps to reduce production time. To program the Arduino UNO microcontroller the Arduino is used. The programming is done strictly in the embedded C language. Arduino is a suite of executable, open source software development tools for the microcontrollers hosted on the Windows platform.Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.One of the difficulties of programming microcontrollers is the limited amount of resources the programmer has to deal with. In personal computers resources such as RAM and processing speed are basically limitless when compared to microcontrollers. In contrast, the code on microcontrollers should be as low on resources as possible

ABOUT ARDUINO COMPILER

GET AN ARDUINO BOARD AND USB CABLE

You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example. (For the Arduino Nano, you'll need an A to Mini-B cable instead.)

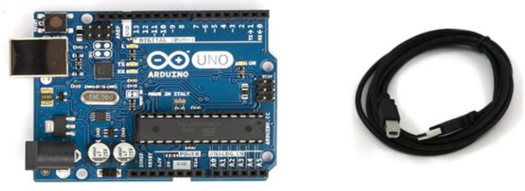
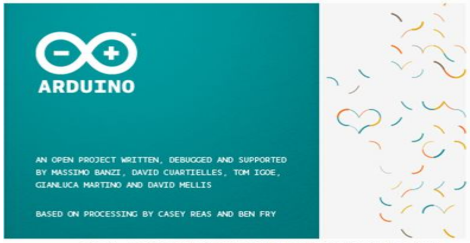


FIG: ARDUINO BOARD AND USB CABLE

CONNECT THE BOARD

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you're using an Arduino Diecimila, you'll need to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it's on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should go on.



AN OPEN PROJECT WRITTEN, DEBUGGED AND SUPPORTED BY MASSIMO BANZI, DAVID CUARTIELLES, TOM IGOE, GIANLUCA MARTINO AND DAVID MELLIS

BASED ON PROCESSING BY CASEY REAS AND BEN FRY

FIG: OPENING THE ARDUINO WINDOW

Open the blink example

Open the LED blink example sketch: File > Examples > 1.Basics > Blink.

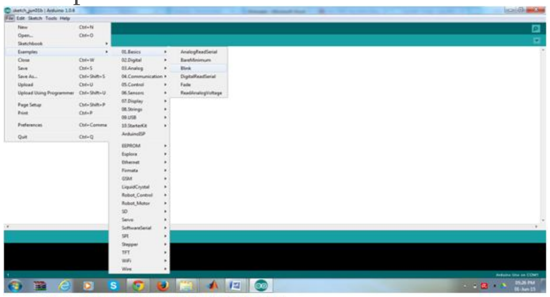
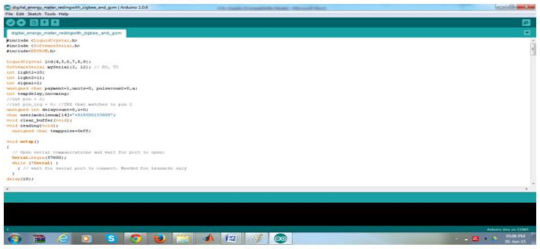


FIG: OPENING BLINK EXAMPLE


digital_energy_meter_redingwith_zigbee_and_gsm | Arduino 1.0.6
File Edit Sketch Tools Help
digital_energy_meter_redingwith_zigbee_and_gsm
Arduino Uno on COM1

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include<EEPROM.h>

LiquidCrystal lcd(4,5,6,7,8,9);
SoftwareSerial mySerial(3, 12); // RX, TX
int light1=10;
int light2=11;
int signal=2;
unsigned char payment=1,units=0, pulsecount=0,a;
int tempdelay,incoming;
//int pin = 2;
//int pin_irq = 0; //IRQ that matches to pin 2
unsigned int delaycount=0,i=0;
char userlmobilenum[14]="+918500153009";
void clear_buffer(void);
void reading(void);
  unsigned char temppulse=0xff;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(57600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
delay(10);
```

FIG: SOURCE CODE WRITTEN IN ARDUINO COMPILER

Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

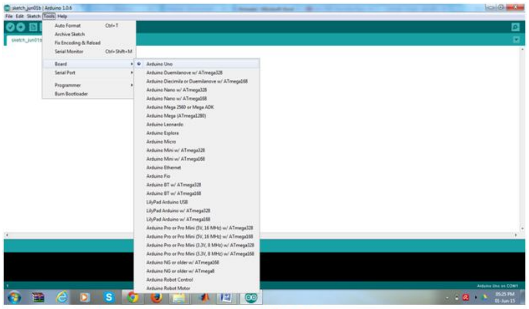


FIG: SELECTING AN ARDUINO UNO

WRITING SKETCHES

Software written using Arduino are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the IDE prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify	Checks your code for errors.
Upload	Compiles your code and uploads it to the Arduino I/O board. See uploading below for details. Note: If you are using an external programmer, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
New	Creates a new sketch.
Open	Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window. Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketch book menu instead.
Save	Saves your sketch.
SerialMonitor	Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive which means only those items relevant to the work currently being carried out are available.

SELECT YOUR SERIAL PORT

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

UPLOAD THE PROGRAM

Before uploading your sketch, you need to select the correct items from the **Tools** > **Board** and **Tools** > **Serial Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be **/dev/ttyUSB0**,**/dev/ttyUSB1** or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **File** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino environment will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar. (*Note:* If you have an Arduino Mini, NG, or other board, you'll need to physically present the reset button on the board immediately before pressing the upload button.)

Compiling sketch...

Arduino Uno on COM1

FIG: COMPILATION UNDER PROCESS

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running.

RESULTS AND DISCUSSIONS

WORKING PROCEDURE

IOT agriculture system based on LORAWAN technology is a very useful and innovative project that helps the farmers to water their fields in a better way. This system uses LORA module for the communication between the farm and the farmer. This system will be placed in the farm field.

The sensors such as LDR and soil moisture sensor are used in this project. The outputs of these sensors are provided to the analog inputs of Arduino. Based on the LDR value, the buzzer will be activated on/off accordingly. The soil moisture sensor senses the moisture level in the field and this value is read by Arduino continuously and if the moisture level is low in the field, Arduino will switch on the motor immediately to water the field. TRIAC is used to control the

AC motor using the input provided by Arduino. This information of the sensor values will be transmitted by Lora module and also displayed on the LCD continuously.

At the receiver side, the sensor values will be received and displayed on the LCD. At the same time, the data will be uploaded onto the web page using Wi-Fi module.

Future Scope:

With IOT we can control the loads the loads with LoRa communication over wifilike AC motor on 4 off,field and moisture level observing,auto on & off water suppy to the field or plants

Reference:-

1.Danco Davcev, Kosta Mitreski, Stefan Trajkovic, Viktor Nikolovski, Nikola Koteli, “IoT agriculture system based on LoRaWAN”, IEEE 2018.

2. Bandur, D.; Jaksic, B.; Bandur, M.; Jovic, S. An Analysis of Energy Efficiency in Wireless Sensor Network (WSNs) applied in smart agriculture. Comput. Electron. Agric. **2019**, 156, 500–507.

3. Muangprathuba, J.; Boonnama, B.; Kajornkasirat, S.; Lekbangpong, N.;Wanichsombat, A.; Nillaor, P. IoT and agriculture data analysis for smart farm. Comput. Electron. Agric. **2019**, 156, 467–474.

4. Goap, A.; Sharma, D.; Shukla, A.; Krishna, C. An IoT based smart irrigation management system using Machine learning and open source technologies. Comput. Electron. Agric. **2018**, 155, 41–49.

5. Munir, M.S.; Bajwa, I.S.; Naeem, M.A.; Ramzan, B. Design and Implementation of an IoT System for Smart Energy Consumption and Smart Irrigation in Tunnel Farming. Energies **2018**, 11, 3427.

6. Tzounis, A.; Katsoulas, N.; Bartzanas, T.; Kittas, C. Internet of Things in agriculture, recent advances and future challenges. Biosyst. Eng. **2017**, 164, 31–48.