

Crop Disease Detection And Control Robot

Yalwar Triveni¹, K. Sudha Rani², S.Mahaboob Basha³

¹P.G. Scholar, ²Assistant Professor, ³Head Of The Department

^{1,2,3} Branch : Embedded Systems

M. Tech

^{1,2,3}Electronics And Communication Engineering Department

^{1,2,3}Geethanjali College Of Engineering And Technology

Email: ¹triveniraghavendragoud@gmail.com, ²komireddy.sudha@gmail.com,

³syedmahaboob45@gmail.com

Abstract

In the proposed system robotics model provides a facility to control the movement of agriculture vehicle. The quality and quantity of agricultural products can reduce by Plant diseases which have produced an enormous post effect scenario. Early pest detection is a major issue dealt with the plantation crops. First step involves in keen and regular observation of plants. Then the diseased plants will be classified and the affected part of the plants images will be acquired using camera. These images are then subjected to pre-processing, transformation and clustering. Then, these images are given as input to the processor, and the processor will compare the images. If the image given is affected image, then an automatic pesticide sprayer is involved to spray the pesticide to the localized area in the leaf. If not, the processors will automatic discard it and the robot will move forward.

INTRODUCTION

India is a land of agriculture. Two-third of population relies upon agriculture for their livelihood. It is the basic foundation of economic development of the country. The agriculture also provides employment opportunities to very large percentage of population. Plant health condition plays a vital role to earn good profit for the farmers. Proper monitoring of plant health is required at different stages of plant growth in order to prevent disease affecting plants.

Existence of pests and disease affect the estimation of crop cultivation and minimizes crop yield substantially. Present day system depends on naked eye observation which is a time consuming process. Automatic detection of plant disease can be adopted to detect plant disease at early stages. Various disease management strategies have been used by farmers at regular intervals in order to prevent plant diseases.

The main aim of the proposed system is to detect plant diseases using Robotics. In most of the plants the disease inception takes place on plant leaves. Hence, in the proposed work we have considered detection of plant disease present on leaves. The discrimination of normal and affected plant leaf can be measured based on variation in temperature, humidity and colour.

LITERATURE SURVEY:

The following papers have been cited during the literature survey to understand the different applications of computer systems in allied areas of the present work carried out.

Mark Seelye et al., 2011 have presented low cost colour sensors for monitoring plant growth in a laboratory. An automated system for measuring plant leaf colour is developed to check plant health status.



Sushma R. Huddar et al., 2012 have presented novel algorithm for segmentation and automatic identification of pests on plants using image processing. The proposed methodology involves reduced computational complexity and aims at pest detection not only in a greenhouse environment but also in a farm environment as well. The whitefly, a bio-aggressor which poses a threat to a multitude of crops, was chosen as the pest of interest in this paper. The algorithm was tested for several whiteflies affecting different leaves and an accuracy of 96% of whitefly detection was achieved.

Murali Krishnan and Jabert.G, 2013 have presented pest control in agricultural plantations using image processing techniques in MATLAB. Images are then subjected to pre-processing, transformation and clustering.

Prof. S. G. Galande, et al., 2015 have presented IoT Implementation for wireless monitoring of agricultural parameters. Wireless system is developed to monitor environmental conditions in agriculture field like temperature, soil pH, soil wet level and humidity beside leaf diseases detection.

Yun Shi et al., 2015 have presented IoT application to monitoring plant diseases and insect pests. IoT technology to percept information, and the role of the IOT technology in agricultural disease and insect pest control, which includes agricultural disease and insect pest monitoring system, collecting disease and insect pest information using sensor nodes, data processing and mining, etc have been described in this paper.

Nimish Gopal, 2016 have presented micro-controller based auto-irrigation and pest detection using image processing. The

technique of image analysis is extensively applied to agriculture science to provide maximum protection to crops which can ultimately lead to better crop management and production.

S. Gavaskar and A. Sumithra, 2017 have presented design and development of pest monitoring system for implementing precision agriculture using IoT. India's most of the farmer grow sugarcane but did not get yielding due to bugs and larvae in sugarcane. In this proposed design system used arduino for monitoring the noise and temperature.

Sai Vivek et al., 2017 have presented arduino based pest control using real time environmental monitoring sensors. This paper strives to develop a robot capable of performing operation of dispensing pest control agents, obstacle avoidance for self-guidance on the field without any user interference and create a sterile environment for the optimum growth of the crops in a real time monitored closed environment.

Oliver Schmittmann et al., 2017 have presented a true-color sensor and suitable evaluation algorithm for plant recognition. The system developed is based on free cascaded and programmable true-colour sensors for real time recognition and identification of individual weed and crop plants using mathematical algorithms and decision models.

Zhang Chuanlei et al., 2017 have presented apple leaf disease identification using genetic algorithm and correlation based feature selection method. A color transformation structure for the input RGB (Red, Green and Blue) image was designed firstly and then RGB model was converted to HSI (Hue, Saturation and Intensity), YUV and gray models. The background was removed and then the disease spot image



was segmented with region growing algorithm (RGA). Finally, the diseases were recognized by SVM classifier.

K.Lakshmi and S.Gayatri, 2017 have presented implementation of IoT with image processing in plant growth monitoring system. This work combines image processing and IoT to monitor the plant and to collect the environmental factors such as humidity and temperature. Plant diseases seriously affect the normal growth of plants, the yield and quality of agricultural products. In recent years, with the dramatic changes in climate, the natural environment of the plant growth has been damaged by pollution, frequent natural disasters, as well as the development of agricultural production.

INTRODUCTION TO EMBEDDED SYSTEMS

Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

Consumer appliances At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today's high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can

carry out many general-purpose tasks such as playing games and word processing.

Office automation: The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation: Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain



the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems

Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Disassemblers (PADs), satellite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

Wireless technologies: Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20th century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

Insemination: Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio

communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

Security: Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in every industrial segment-consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

Finance: Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever we go, we can

see, or at least feel, the work of an embedded system.

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.

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.

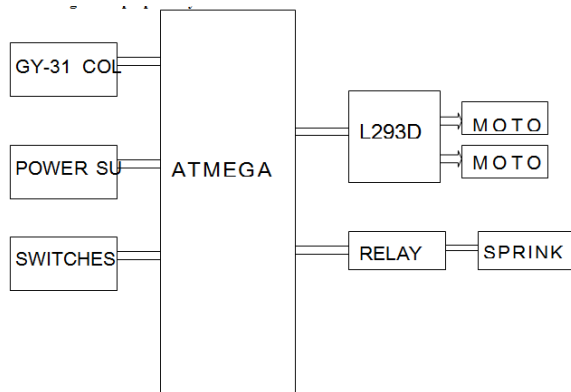
A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O 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, ROM, 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

Block Diagram of the Project

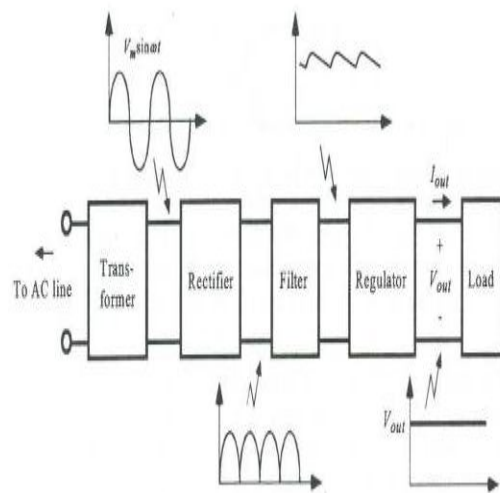
The block diagram of the design is as shown in the figure below. The brief description about block diagram is given below.

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.



Components of a regulated 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.

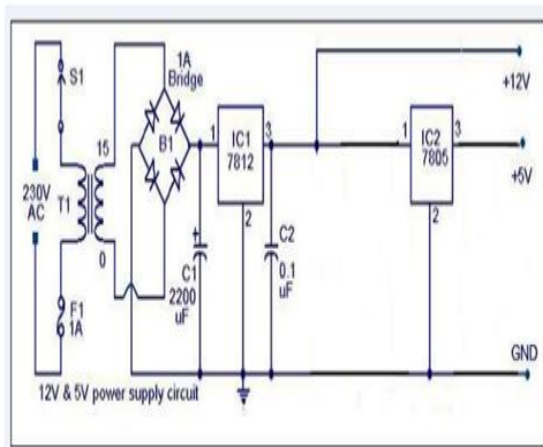


Fig: Power supply circuit diagram

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.

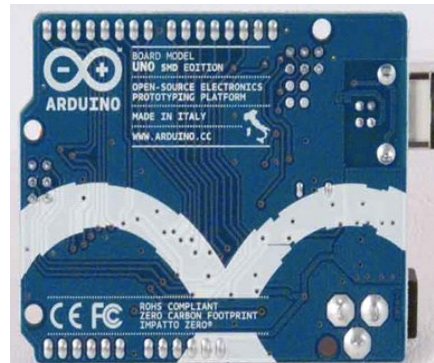


FIG: ARDUINO UNO

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.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a 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, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The

ATmega16 U2/8U2 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 (Windows) or the DFU programmer (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).

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

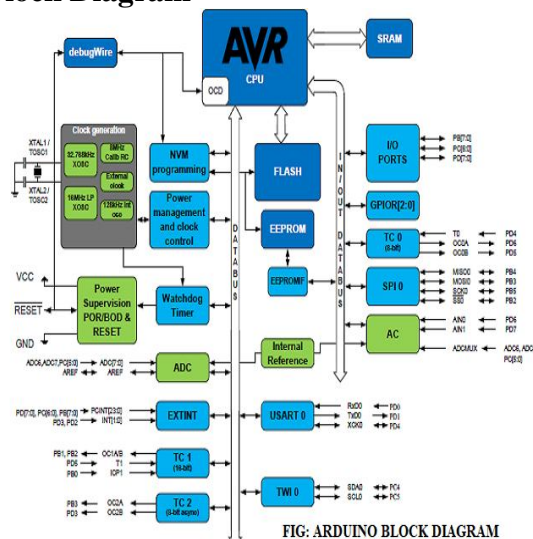


FIG: ARDUINO BLOCK DIAGRAM

Pin Configurations Pin-out

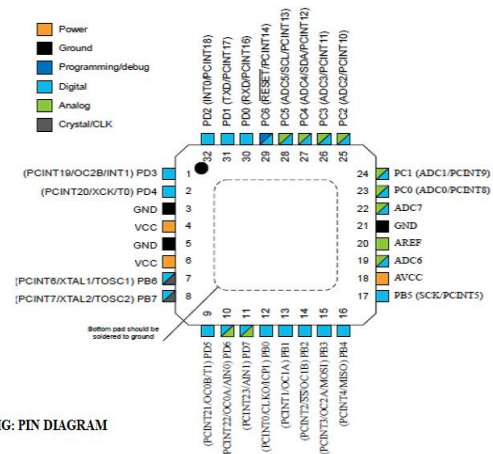


FIG: PIN DIAGRAM

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 unprogrammed, 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 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.

TYPES OF ARDUINO BOARDS:-

Arduino UNO.

- Arduino MEGA.
- Arduino MINI.
- Arduino NANO
- Arduino DUE.
- Arduino YUN.
- Arduino Lily pad.
- Arduino Duemilanova.

Apart from this there are many more boards that can be used. As it's open source instead of Arduino you can also find, Freeduino, Arkduino etc. available in the market. Selection of Board should be done according to the application.

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
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 14 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.

GY-31 TCS3200 Color Sensor Module

This module is based on the TCS3200 chip, an upgraded version of the TCS230, which is a programmable color light to frequency

converter, making it ideal for adding color sensing capabilities to your new project.



Fig: GY-31 Colour Sensor Module

Features:

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package

Pin description:

Pin	Pin name	Description
1	GND	Power supply ground
2	OUT	Output frequency
3	S2	Photodiode type selection inputs
4	S3	Photodiode type selection inputs
5	VCC	Supply Voltage. 2.7-5v
6	VCC	Supply Voltage. 2.7-5v
7	S1	Output frequency scaling selection inputs
8	S0	Output frequency scaling selection inputs
9	LED	LED CONTROL 1 : LED ON , 0 : LED OFF
10	GND	Power supply ground

Description:

The TCS3200 and TCS3210 are programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square



wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins.

Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line. In the TCS3200, the light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. In the TCS3210, the light-to-frequency converter reads a 4 x 6 array of photodiodes. Six photodiodes have blue filters, 6 photodiodes have green filters, 6 photodiodes have red filters, and 6 photodiodes are clear with no filters.

The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of the same color are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 μm x 110 μm in size and are on 134- μm centers.

Switch

Switch is an electrical component which can make or break electrical circuit automatically or manually. Switch is mainly works with ON (open) and OFF (closed) mechanism. Numerous circuits hold switches that control how the circuit works or actuate different characteristics of the circuit.

The classification of switches depends on the connection they make. Two vital components that confirm what sorts of connections a switch makes are pole and

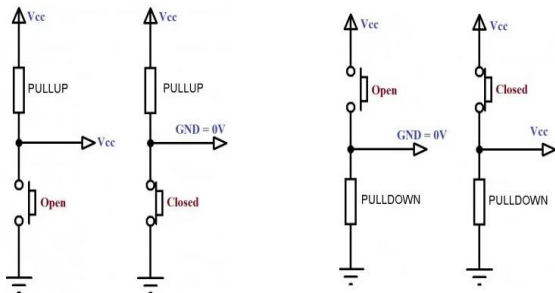
throw. These are classified on based the connections they make.

The terms pole and throw are also used to describe switch contact variations. The number of "poles" is the number of separate circuits which are controlled by a switch. The number of "throws" is the number of separate positions that the switch can adopt. A single-throw switch has one pair of contacts that can either be closed or open. A double-throw switch has a contact that can be connected to either of two other contacts; a triple-throw has a contact which can be connected to one of three other contacts, etc.

Pole: The amount of circuits controlled by the switch is indicated by poles. Single pole (SP) switch controls only one electrical circuit. Double pole (DP) switch controls two independent circuits.

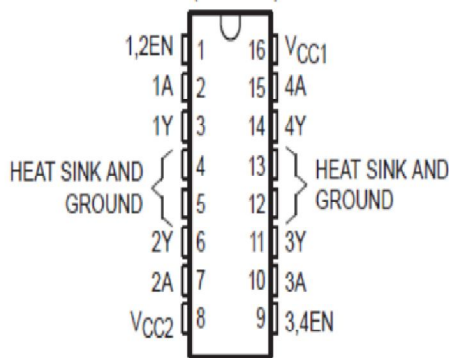
Throw: The number of throws indicates how many different output connections every switch pole can connect its input. A single throw (ST) switch is a simple on/off switch. When the switch is ON, the two terminals of switch are connected and current flows between them. When the switch is OFF the terminals are not connected, so current does not flow.

A limit switch is a switch operated by the motion of a machine part or presence of an object. A limit switch is an electromechanical device that consists of an actuator mechanically linked to a set of contacts. When an object comes into contact with the actuator, the device operates the contacts to make or break an electrical connection. A PULL-UP or PULL-DOWN resistor is to connected while interfacing switch to any controller.



If we haven't use PULL-UP or PULL-DOWN resistors, there will be an UNDETERMINED STATE (neither LOW nor HIGH) when the switch is OPEN.

L293D- CURRENT DRIVER CHIP



Pin diagram FEATURES

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

DESCRIPTION

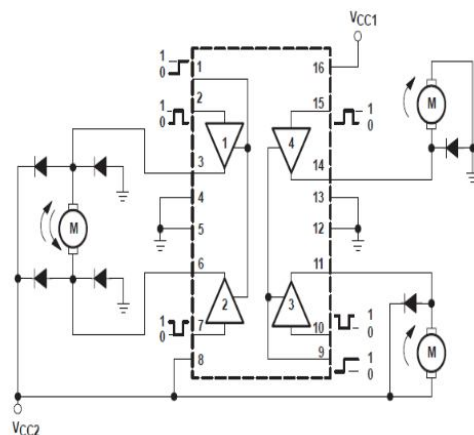
The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The

L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0 to 70 degree Celsius.

BLOCK DIAGRAM



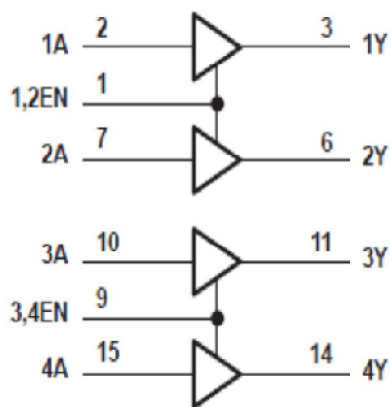
FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

† In the thermal shutdown mode, the output is
in the high-impedance state, regardless of
the input levels.

LOGIC DIAGRAM



This chip contains 4 enable pins. Each enable pin corresponds to 2 inputs. Based on the input values given, the device connected to this IC works accordingly.

DC Motor

DC motors are configured in many types and sizes, including brush less, servo, and gear motor types. A motor consists of a rotor and a permanent magnetic field stator. The magnetic field is maintained using either permanent magnets or electromagnetic windings. DC motors are most commonly used in variable speed and torque.

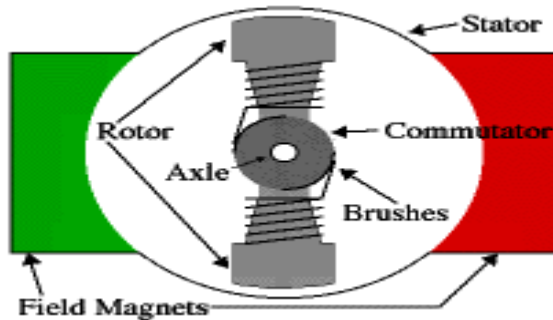
Motion and controls cover a wide range of components that in some way are used to generate and/or control motion. Areas within this category include bearings and bushings, clutches and brakes, controls and drives, drive components, encoders and resolvers, Integrated motion control, limit switches, linear actuators, linear and rotary

motion components, linear position sensing, motors (both AC and DC motors), orientation position sensing, pneumatics and pneumatic components, positioning stages, slides and guides, power transmission (mechanical), seals, slip rings, solenoids, springs.

Motors are the devices that provide the actual speed and torque in a drive system. This family includes AC motor types (single and multiphase motors, universal, servo motors, induction, synchronous, and gear motor) and DC motors (brush less, servo motor, and gear motor) as well as linear, stepper and air motors, and motor contactors and starters.

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

Let's start by looking at a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

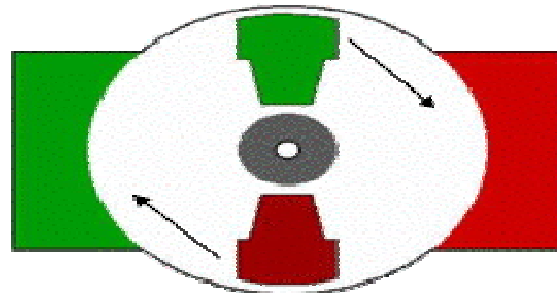


Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beemers will see), the external magnetic field is produced by high-strength permanent magnets¹. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

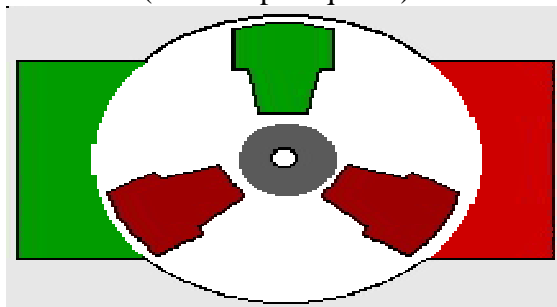
The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, and driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can

imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque" ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

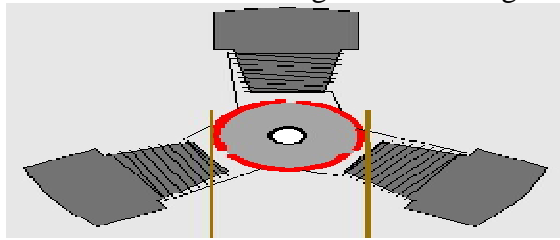


So since most small DC motors are of a three-pole design, let's tinker with the workings of one via an interactive animation (JavaScript required):



You'll notice a few things from this -- namely, one pole is fully energized at a time (but two others are "partially" energized). As each brush transitions from one commutator contact to the next, one coil's field will rapidly collapse, as the next coil's field will rapidly charge up (this occurs within a few microsecond). We'll see more about the effects of this later, but in the

meantime you can see that this is a direct result of the coil windings' series wiring:



There's probably no better way to see how an average dc motor is put together, than by just opening one up. Unfortunately this is tedious work, as well as requiring the destruction of a perfectly good motor.

Relay

A relay is an electrically controllable switch widely used in industrial controls, automobiles and appliances. The relay allows the isolation of two separate sections of a system with two different voltage sources i.e., a small amount of voltage/current on one side can handle a large amount of voltage/current on the other side but there is no chance that these two voltages mix up.

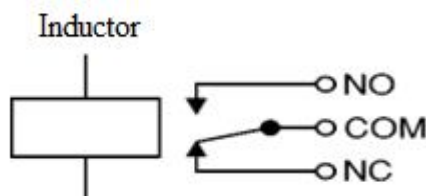


Fig: Circuit symbol of a relay

Operation:

When the current flows through the coil, magnetic field is created around the coil i.e., the coil is energized. This causes the armature to be attracted to the coil. The armature's contact acts like a switch and closes or opens the circuit. When the coil is not energized, a spring pulls the armature to its normal state of open or closed. There are all types of relays for all kinds of applications.

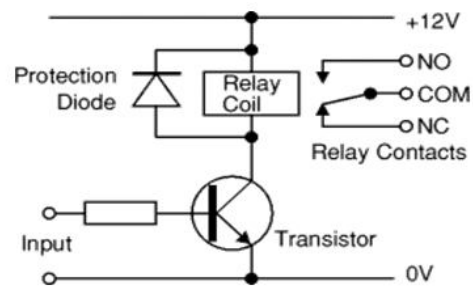


Fig: Relay Operation and use of protection diodes

Transistors and ICs must be protected from the brief high voltage 'spike' produced when the relay coil is switched off. The above diagram shows how a signal diode (eg 1N4148) is connected across the relay coil to provide this protection. The diode is connected 'backwards' so that it will normally not conduct. Conduction occurs only when the relay coil is switched off, at this moment the current tries to flow continuously through the coil and it is safely diverted through the diode. Without the diode no current could flow and the coil would produce a damaging high voltage 'spike' in its attempt to keep the current flowing.

In choosing a relay, the following characteristics need to be considered:

1. The contacts can be normally open (NO) or normally closed (NC). In the NC type, the contacts are closed when the coil is not energized. In the NO type, the contacts are closed when the coil is energized.
2. There can be one or more contacts. i.e., different types like SPST (single pole single throw), SPDT (single pole double throw) and DPDT (double pole double throw) relays.
3. The voltage and current required to energize the coil. The voltage can vary from a few volts to 50 volts, while the current can be from a few milliamps to 20milliamps. The relay has a minimum voltage, below which the coil will not be



energized. This minimum voltage is called the “pull-in” voltage.

4. The minimum DC/AC voltage and current that can be handled by the contacts. This is in the range of a few volts to hundreds of volts, while the current can be from a few amps to 40A or more, depending on the relay.

Driving a Relay:

An SPDT relay consists of five pins, two for the magnetic coil, one as the common terminal and the last pins as normally connected pin and normally closed pin. When the current flows through this coil, the coil gets energized. Initially when the coil is not energized, there will be a connection between the common terminal and normally closed pin. But when the coil is energized, this connection breaks and a new connection between the common terminal and normally open pin will be established. Thus when there is an input from the microcontroller to the relay, the relay will be switched on. Thus when the relay is on, it can drive the loads connected between the common terminal and normally open pin. Therefore, the relay takes 5V from the microcontroller and drives the loads which consume high currents. Thus the relay acts as an isolation device.

Digital systems and microcontroller pins lack sufficient current to drive the relay. While the relay’s coil needs around 10milli amps to be energized, the microcontroller’s pin can provide a maximum of 1-2milli amps current. For this reason, a driver such as ULN2003 or a power transistor is placed in between the microcontroller and the relay. In order to operate more than one relay, ULN2003 can be connected between relay and microcontroller.

Applications:

1. Home appliances Air conditioner, heater, etc.

2. Office machines PPC, Facsimile, etc
3. Vending machines

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 the following sections.

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.)

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.

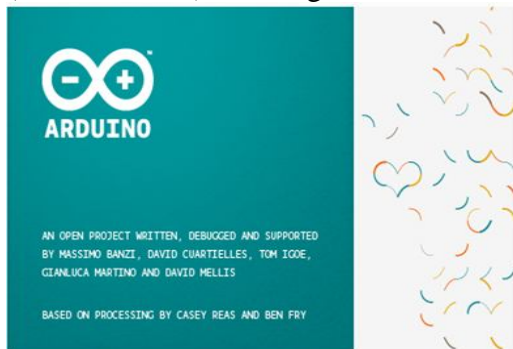


FIG: OPENING THE ARDUINO WINDOW

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

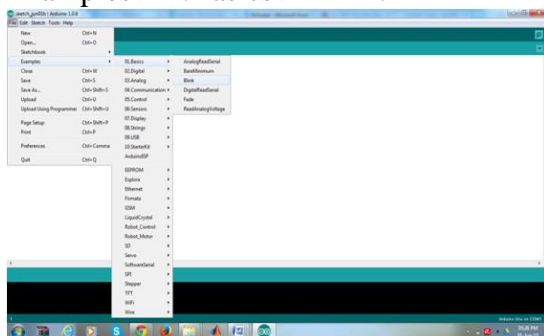


FIG: OPENING BLINK EXAMPLE

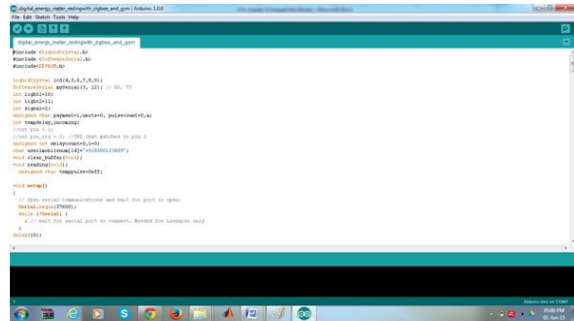


FIG: SOURCE CODE WRITTEN IN ARDUINO COMPILER

Select your board

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

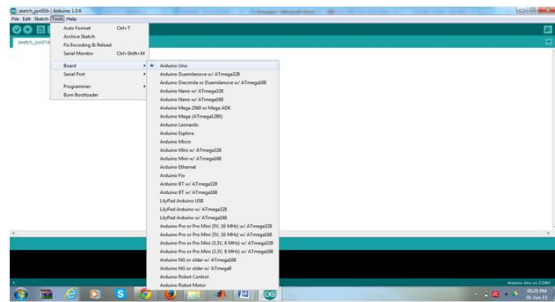


FIG: SELECTING AN ARDUINO UNO

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

1.1.1. Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

- 2.
3. 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 boot loader, 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 boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader 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.)



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

In this model, an automated system has been developed to determine whether the plant is normal or diseased. The normal growth of the plants, yield and quality of agricultural products is seriously affected by plant disease. This project is an automated system that detects the presence of disease in the plants. An automated disease detection system is developed using sensors like colour sensor based on variation in plant leaf health condition. The values based on colour parameters are used to identify presence of plant disease.

Changes in the color of plant tissue are a common symptom of plant disease. Often these color changes are brought about by the yellowing of normal green tissue due to the destruction of chlorophyll or a failure to form chlorophyll. Such repression of leaf color may be complete or partial. The color sensor senses the color of the leaf under consideration which is another parameter that is being used to determine whether the leaf is either diseased or healthy.

We have used TCS3200 RGB color sensor. The color values that are recorded for the leaf are then sent to the Arduino board for analysis. Later the obtained values of RGB are compared with the threshold value in dataset to determine whether the leaf is healthy or diseased.

The Robot arrangement is made in this project. The robot will be moving around the crop field based on the inputs given from the switches. The Arduino continuously reads the values from the colour sensor and if it detects that the leaf is diseased, the pesticide will be sprayed on the leaf to save the leaf.

Reference:-

- [1] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 34-41, 2005.
- [2] F. Benzi, N. Anglani, E. Bassi, and L. Frosini, "Electricity Smart Meters Interfacing the Households," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4487-4494, 2011.
- [3] A. Zaballos, A. Vallejo, M. Majoral, and J. M. Selga, "Survey and Performance Comparison of AMR Over PLC Standards," *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 604-613, 2009.
- [4] R. Rashed Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on Advanced Metering Infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473-484, 2014
- [5] A. M. Vega, F. Santamaria, and E. Rivas, "Modeling for home electric energy management: A review," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 948-959, 2015.
- [6] E. Taktak and I. B. Rodriguez, "Energy Consumption Adaptation Approach for Smart Buildings," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 1370-1377.
- [7] M. Beaudin and H. Zareipour, "Home energy management systems: A review of modelling and complexity," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 318-335, 2015.
- [8] K. Athira, A. R. Devidas, M. V. Ramesh, and V. P. Rangan, "User centered energy management scheme for smart buildings," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1296-1302.
- [9] H. G. R. Tan, C. H. Lee, and V. H. Mok, "Automatic power meter reading system



- using GSM network,” in 2007 International Power Engineering Conference (IPEC 2007), 2007, pp. 465-469.
- [10] M. Wasi-ur-Rahman, M. T. Rahman, T. H. Khan, and S. M. L. Kabir, “Design of an intelligent SMS based remote metering system,” in 2009 International Conference on Information and Automation, 2009, pp. 1040-1043.
- [11] L. Labib, M. Billah, G. M. S. M. Rana, M. N. Sadat, M. G. Kibria, and M. R. Islam, “Design and implementation of low-cost universal smart energy meter with demand side load management,” IET Generation, Transmission & Distribution, vol. 11, no. 16, pp. 3938-3945, 2017.
- [12] A. H. Primicanta, N. Mohd Yunus, and M. Awan, “ZigBee-GSM based Automatic Meter Reading system,” in 2010 International Conference on Intelligent and Advanced Systems, 2010, pp. 1-5.
- [13] M. A. Prada, P. Reguera, S. Alonso, A. Morán, J. J. Fuertes, and M. Domínguez, “Communication with resource-constrained devices through MQTT for control education,” IFAC-PapersOnLine, vol. 49, no. 6, pp. 150-155, 2016