A

Thesis

Report on

SMART CROP PROTECTION SYSTEM FROM ANIMALS BY USING ARDUINO

Submitted to

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING

By

B. LEELA MANI 18P11A0411

Under the Esteemed by

Dr. K. SOUNDARARAJAN GARU

Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE

(AUTONOMOUS)

(Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi, Affiliated to JNTU Anantapur) Renigunta road, Tirupati-517506, Andhra Pradesh, India

2018-2022

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE

(AUTONOMOUS)

Department of Electronics & Communication Engineering

Certificate

This is to certify that the project work entitled **“SMART CROP PROTECTION SYSTEM FROM ANIMALS”** is a bonafide work done by **B.LEELAMANI(18P11A0411)**in the Department of **“ELECTRONICS & COMMUNICATION ENGINEERING”**, and submitted to **Chadalawada Ramanamma Engineering College (Autonomous), Tirupati** is a project work carried out by them under my guidance during the academic year 2020-2021.

GUIDE	**HEAD**
Dr. K. SOUNDARARAJAN GARU	**Dr. P. KRISHNA MURTHY M. Tech, Ph.D.**
Professor	Associate Professor & Head
Department of ECE	Department of ECE

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We are hereby declared that the project work on **“SMART CROP PROTECTION SYSTEM FROM ANIMALS**” done by us under guidance by **Dr. K. SOUNDARARAJAN garu in CHADALAWADA RAMANAMMA ENGINEERING COLLEGE (Autonomous)** is submitted in partial fulfillment of the requirements of the requirements of the award of Degree of Bachelor of Technology**.**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our project report submission.

Signature of the Students

B. LEELAMANI **18P11A0411**

ACKNOWLEDGEMENT

A grateful thanks to **Dr. CHADALAWADA KRISHNAMURTHY**, Chairman, C.V.S. Krishna Murthy Teja Charities for providing education in this esteemed Institution.

We express our gratitude to our principal **Dr. P. SANJEEVA RAYUDU**, **MTech, Ph.D.** Principal, Chadalawada Ramanamma Engineering College, Tirupati, for providing all the facilities and supporting us to complete this project work.

Our sincere thanks **to Dr. P. KRISHNA MURTHY**, **MTech, Ph.D.** Associate Professor and Head of the Department of Electronics & Communication Engineering, Chadalawada Ramanamma Engineering College, Tirupati, for providing infrastructural facilities to complete our project work.

We are greatly indebted to our guide, **Dr. K. SOUNDARARAJAN** garu Professor in Electronics & Communication Engineering, Chadalawada Ramanamma Engineering College, Tirupati, who encourages us in all spheres of activities, constant encouragement and keen interest enriched to complete this work.

We extend our thanks to all the Teaching staff of the Department of E.C.E. for their support, encouragement and inspired us. We also thank the non-Teaching staff of the Department of E.C.E. for being helpful in many ways in successful completion of our work.

The chain of our gratitude would be definitely incomplete, if we forget to thank all my friends of Chadalawada Ramanamma Engineering College, Tirupati for their constant support and encouragement. Finally, we thank all those who helped us directly or indirectly in successful completion of our project work.

B. LEELA MANI **18P11A0411**

ABSTRACT

Generally, in many fields the crops are spoiled by animals like cows, buffaloes, birds, pigs etc. It is not possible for the farmers to take care of his crops all the time from animals. So here we design a project about “smart crop protection system from animals”. This project is based on Arduino UNO micro controller. This system uses a motion sensor to detect wild animals approaching near the fields. In such a case the sensor signals will approach the microcontroller to take action. Then the microcontroller sound an alarm which makes the animals move away from the fields and the microcontroller activates LED simultaneously. This ensures complete safety of crops from animals and reduces the farmers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Short form	**Full form**
PCB	Printed circuit board
IDII	Interaction Design Institute Ivrea
IDE	Integrated development Environment
AT mega	Atmel Mega AVR family
AVR	Automatic Voltage Regulator
RISC	Reduced Instruction Set Computer
ARM	Adjustable-Rate Mortgage
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
FTDI	Future Technology Devices International
ICSP	In Circuit Serial Programming
ISP	In System Programming
GNU	General Public License

CLI	Command Line Interface
Arduino RS232	Arduino Recommended standard 232
SMD R3	Surface Mounted Device R3
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
IOREF	I\O Voltage Reference
AREF	Analog Reference
TWI	Two- Wire Interference
ALU	Arithmetic Logic Unit
SRAM	Static Random-Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
MCU	Micro Chip AT mega 328P
BOD	Brown Out Detection
ADC	Analog to Digital Convertor
WDT	Watch Dog Timer
SMCR	Sender Message, Channel and Receiver
TIMSK	Timer/Counter Interrupt Mask Register
SREG	Status Register

CHAPTER 1

INTRODUCTION TO ARDUINO

1.1 HISTORY OF ARDUINO

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a basic Stamp microcontroller at a cost of $50. In 2003 Hernando Barragan created the development program. Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Rea's is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2005, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, extended Wiring by adding support for the cheaper ATmega8 microcontroller. The new project, forked from Wiring, was called ARDUINO.

At the end of 2008, Gianluca Martino's company, Smart Projects, registered the Arduino trademark in Italy and kept this a secret from the other co-founders for about two years. This was revealed when the Arduino company tried to register the at trademark in other areas of the world (they originally registered only in the US), and discovered that it was already registered in Italy. Negotiations with Martino and his firm to bring the trademark under control of the original Arduino company failed. In 2014, Smart Projects began refusing to pay royalties. They then appointed a new CEO, Federico Musto, who renamed the company Arduino SRL and created the website arduino.org, copying the graphics and layout of the original arduino.cc. This resulted in a rift in the Arduino development team.

In October 2017, Arduino announced its partnership with ARM Holdings (ARM). The announcement said, in part, "ARM recognized independence as a core value of Arduino ... without any lock-in with the ARM architecture.” Arduino intends to continue to work with all technology vendors and architectures. Under Volante’s guidance, the company started growing again and releasing new designs. The Genuine trademark was dismissed and all products were branded again with the Arduino name. As of February 2020, the Arduino community included about 30 million active users based on the IDE downloads.

In August 2018, Arduino announced its new open-source command line tool which can be used as a replacement of the IDE to program the boards from a shell. In February 2019, Arduino announced its IoT Cloud service as an extension of the Create online environment.

1.2 HARDWARE OF ARDUINO

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -Arduino.

Most Arduino boards consist of an Atmel 8-bit AVR Microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lilypad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino Uno is the Opti boot bootloader.[31] Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Arduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

1.3 SOFTWARE OF ARDUINO

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

1.3.1 IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

1.3.2 PRO IDE

On October 18, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, autocompletion support, and Git integration. The application frontend is based on the Eclipse Theia Open-Source IDE. The main features available in the alpha release.

- Modern, fully featured development environment
- Dual Mode, Classic Mode (identical to the Classic Arduino IDE) and Pro Mode (File System view)
- New Board Manager
- New Library Manager
- Board List
- Basic Auto-Completion (Arm targets only)
- Git Integration
- Serial Monitor
- Dark Mode

1.3.3 SKETCH

A sketch is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension in Arduino Software (IDE) pre-1.0 saved sketches with the extension. pde.

A minimal Arduino C/C++ program consists of only two functions:

- `setup ()`: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function `main ()`.
- `loop ()`: After `setup ()` function exits (ends), the `loop()` function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function `while (1)`.

1.3.4 THREADING

There is a Xinu OS port for the atmega328p (Arduino Uno and others with the same chip), which includes most of the basic features. The source code of this version is freely available.

There is also a threading tool, named Protothreads. Protothreads are described as "... extremely lightweight stack less threads designed for severely memory constrained systems, such as small embedded systems or wireless sensor network nodes. Protothreads provide linear code execution for event-driven systems implemented in C. Protothreads can be used with or without an underlying operating system."

Protothread Ing utilizes C switch () statement in a non-obvious way that is similar to Duff's device. The following is skeleton code for protothreads.

1.4 LIST OF AURDUINO BOARDS

1.4.1 OFFICIAL BOARDS

The original Arduino hardware was manufactured by the Italian company Smart Projects. Some Arduino-branded boards have been designed by the American companies Spark Fun Electronics and Adafruit Industries. As of 2016, 17 versions of the Arduino hardware have been commercially produced.

Fig1.1:Arduino RS232 Fig1.2:Arduino Diecimila

Fig1.3: Arduino Duemilanove (rev 2009b) Fig1.4:Arduino Uno R2

Fig1.5: Arduino Uno SMD R3

Fig1.6: Arduino micro(AtMega 32U4)

Fig1.7: Arduino pro micro (AtMega32U4)

Fig1.8: Arduino Pro

(No USB)

Fig1.9: Arduino Mega Fig1.10: Arduino Nano (DIP-

30 footprint)

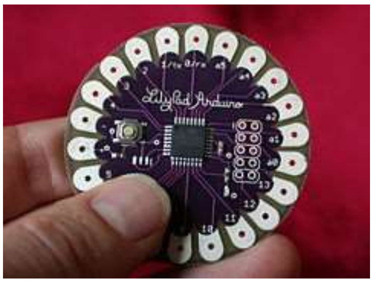


Fig 1.11: Arduino Lilypad 00 Fig 1.12: Arduino

Robot

(rev 2007) (No USB)

Fig 1.13: Arduino Esplora

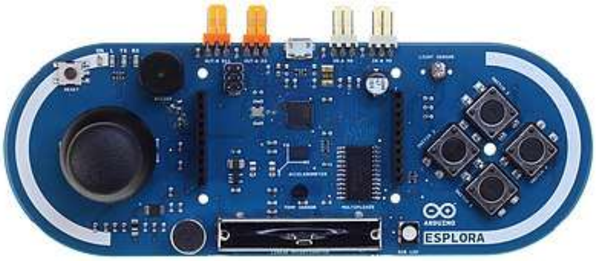


Fig 1.14: Arduino Ethernet (AVR + W5100)

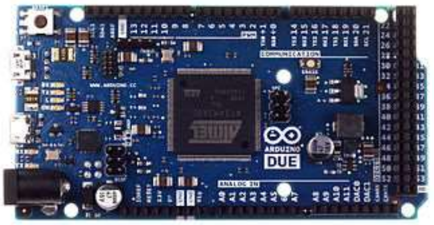


Fig 1.15: Arduino Yún

Fig1.16: Arduino Due (ARM Cortex-M3 core)

1.4.2 COMPATIBLE BOARDS

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education,[34] to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibilities. Arduino-compatible

R3 Uno board made in China with no Arduino logo, but with identical markings, including "Made in Italy".

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Crop damage caused by animal attacks is one of the major threats in reducing the crop yield. Due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most conflicts antagonizing human wildlife relationships. There is an increasing damage of vineyards and farm lands that have resulted in a huge drop in production and about 1000 road accidents are also caused by these wild animals annually. Due to over population, it occurs a deforestation this results in shortage of food, water and shelter in forest areas. So, animals interference in residential areas is increasing day by day which affects human life and property causes human animal conflict but as per nature's rule every living creature on this earth has important role in eco-system. Elephants and other animals coming in to contact with humans, impact negatively in various means such as by depredation of crops, damaging grain stores, water supplies, houses and other assets, injuring and death of humans The current methods used to counter this problem include the use of electrified welded mesh fences. Other traditional methods applied by farmers include the use of Hell kites, Balloons, Shot/Gas guns, String & stone, etc. These solutions are often cruel and ineffective. They also require a vast amount of installation and maintenance cost and some of the methods have environmental pollution effect on both humans and animals. The chemical products used to prevent these animal attacks have an application cost per hectare and their effectiveness is dependent on weather condition, as rain may cause a dilution effect. Farmers in India face serious threats from which one is damage by animals resulting in lower yields. Traditional methods followed by farmers are not that effective and it is not feasible to hire guards to keep an eye on crops and prevent wild animals. Since safety of both human and animal is equally vital. So, animal detection system is necessary in farm areas. Our method is based on an animal friendly ultrasounds generator, which does not produce physical or biological harm to the animals nor sounds harmful to humans. The aim of the project is to address the problem of crop vandalization by wild animals. The project aims to provide an effective solution to this problem, so that the economic losses incurred by our farmers are minimized and they have a good crop yield.

2.2 LITERATURE REVIEW

This paper motive to designing and executing the superior improvement in embedded device for Crops in farms are over and over ravaged with the aid of nearby animals like buffaloes, cows, goats, birds, and fireplace etc. This results in huge losses for the farmers. It is now not feasible for farmers to barricade complete fields or precede field 24 hours and protect it. Therefore, here we present computerized crop safety system from animals and fire. This is a Arduino Uno primarily based device the use of microcontroller. This technique makes use of a motion sensor to discover wild animals drawing near the sphere and smoke sensor to discover the hearth. In such a case the sensor alerts the microcontroller to require action. The microcontroller now sounds an alarm to woo the animals away from the sector further as sends SMS to the farmer and makes call, in order that farmer may fathom the difficulty and come to the spot just in case the animals don’t recede by the alarm. If there's a smoke, it immediately turns

ON the motor. This provides us entire safety of plants from animals and from fireplace for this reason protecting the farmer's loss.

CHAPTER 3

INTRODUCTION OF ATMEGA328P

CHAPTER 3.1: INTRODUCTION OF ARDUINO UNO ATMEGA328P

Arduino Uno is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.The first Arduino project was started in Interaction Design Institute Ivrea in 2003 by David cuartillas and Massimo Banzi with the intention of providing a cheap and flexible way for students and professionals for controlling a number of devices in the real World. The current version of Arduino Uno comes with a USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output. This board comes with all the features required to run the controller and can be directly connected to the computer through a USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems; however, Windows is preferable to use. Programming languages like C and C++ are used in IDE. Apart from USB, a battery or AC to DC adopter can also be used to power the board. Uno boards don't come with FTDI USB to Serial driver chip. There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with the Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB. Arduino Uno comes with a USB interface i.e.Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

3.2 pin description

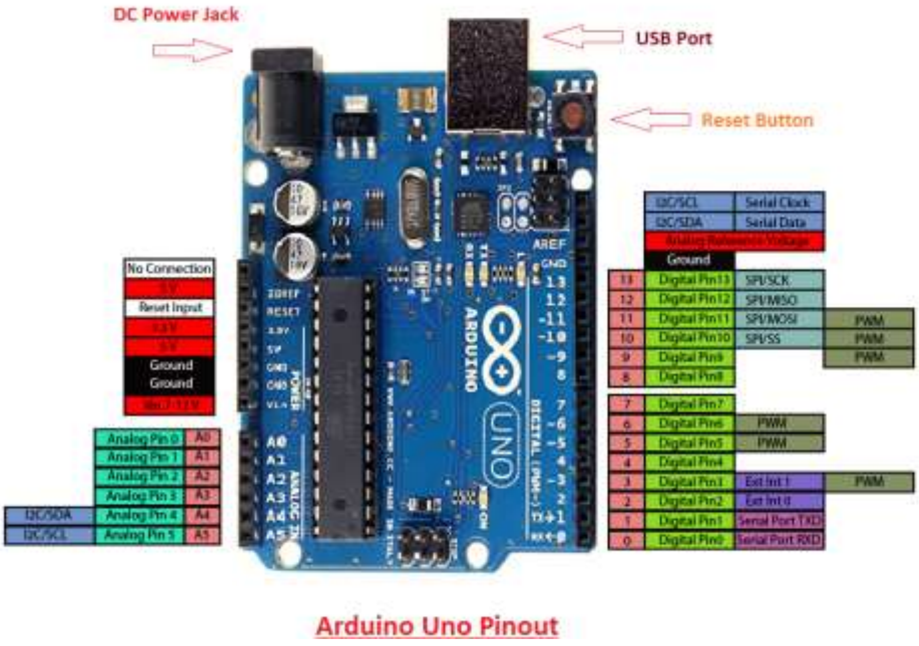


Fig 3.1: Arduino uno pinout

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

- **LED.** Arduino Uno comes with a built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.
- **Vin.** It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through a power jack, it can be accessed through this pin.
- **5V.** This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e USB, Vin pin of the board or DC power jack.
- USB supports voltage around 5V while Vin and Power Jack support a voltage range between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins.
- **Reset.** This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.
- **IOREF.** This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then selects the proper power source.
- **PWM.** PWM is provided by 3,5,6,9,10, 11pins. These pins are configured to provided 8-bit output PWM.
- **SPI.** It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of the SPI library.
- **AREF.** It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.
- **TWI.** It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

- **Serial Communication.** Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).
- Rx pin is used to receive data while Tx pin is used to transmit data.
- **External Interrupts.** Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

3.3 BLOCK DIAGRAM OF ARDUINO UNO ATMEGA328P

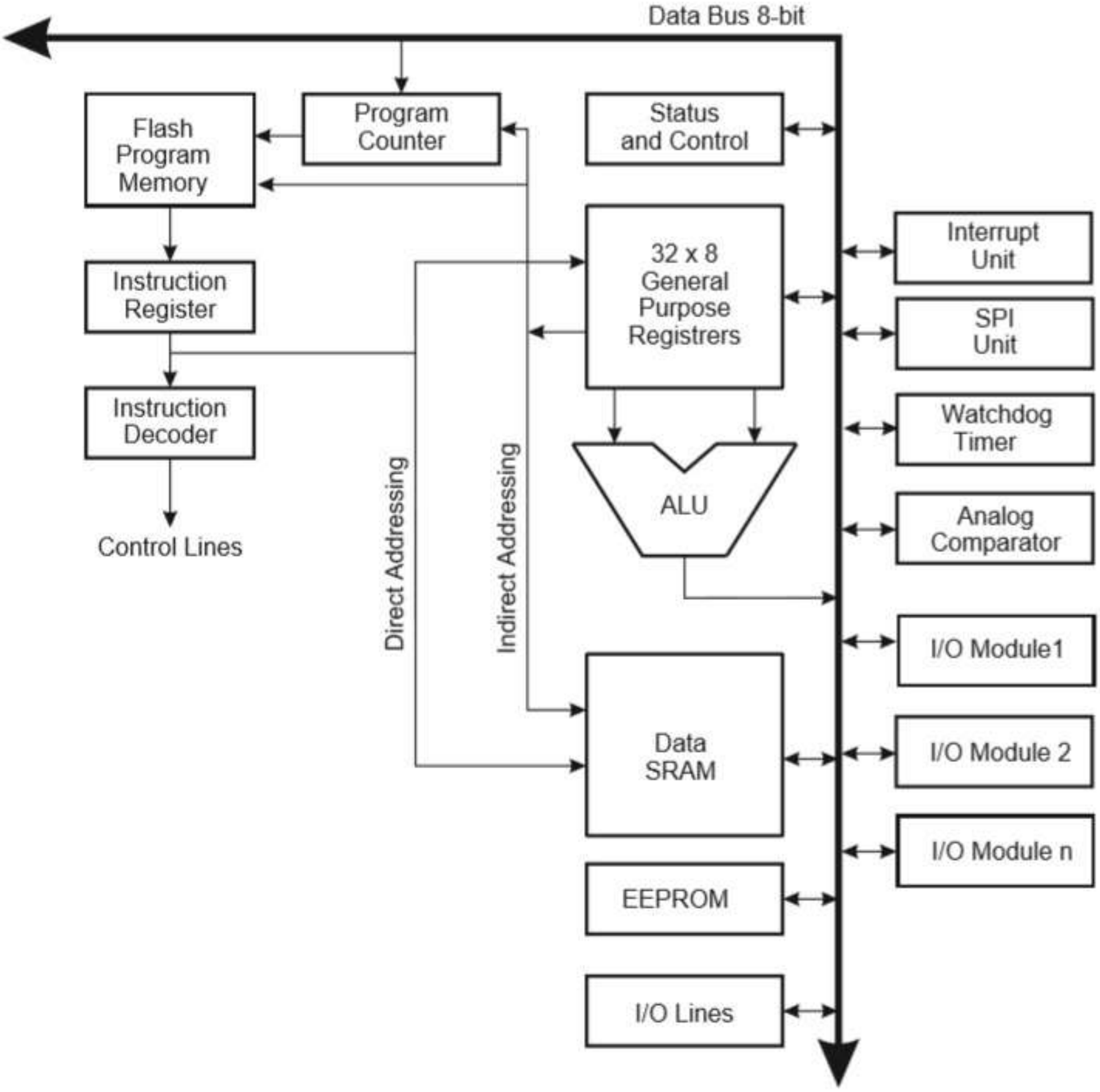


Fig 3.2: block diagram of ATMEGA328P

The basic working of CPU of ATmega328:

1. The data is uploaded in serial via the port (being uploaded from the computer's Arduino IDE). The data is decoded and then the instructions are sent to **instruction register** and it decodes the instructions on the same clock pulse.

2. On the next clock pulse the next set of instructions are loaded in instruction register.

3. In general purpose registers the registers are of 8-bit but there are 3 16-bit registers also.

a. 8-bit registers are used to store data for normal calculations and results.

b. 16-bit registers are used to store data of timer counter in 2 different register. Eg,X-low & X-high. They are fast, and are used to store specific hardware functions.

4. EEPROM stores data permanently even if the power is cut out. Programming inside EEPROM is slow.

5. Interrupt Unit checks whether there is an interrupt for the execution of instruction to be executed in ISR (Interrupt Service Routine).

6. Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as Camera, Display, SD cards, etc. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.

7. Watchdog timer is used to detect and recover from MCU malfunctioning.

8. Analog comparator compares the input values on the positive and negative pin, when the value of positive pin is higher the output is set.

9. Status and control are used to control the flow of execution of commands by checking other blocks inside the CPU at regular intervals.

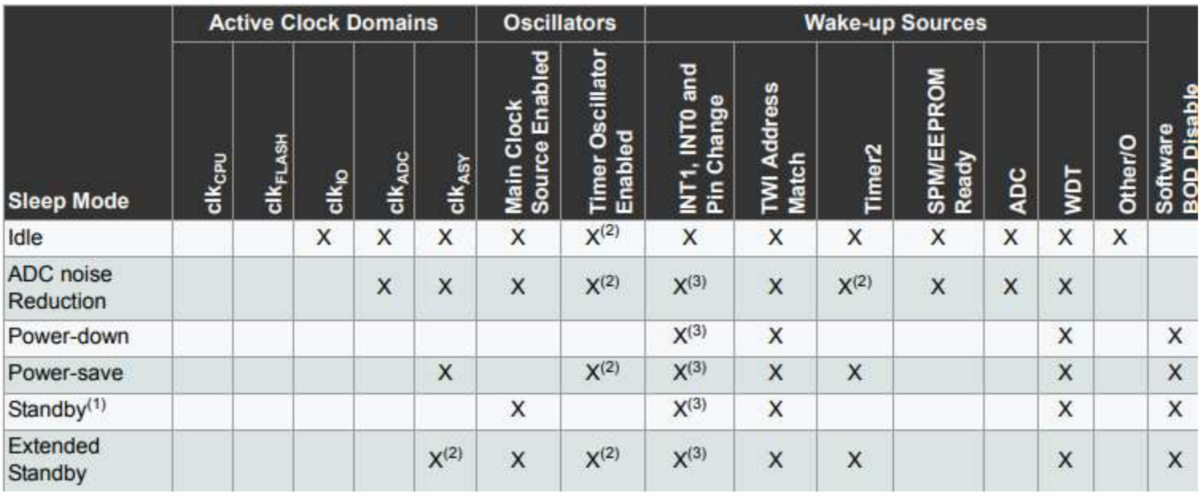
10. ALU (Arithmetic and Logical unit) The high performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations b/w general purpose registers are executed. The ALU operations are divided into 3 main categories – arithmetic, logical and bit-function.

11. I/O pins the digital inputs and outputs (digital I/O) on the Arduino are what allow you to connect the Arduino sensors, actuators, and other ICs. Learning how to use them will allow you to use the Arduino to do some really useful things, such as reading switch inputs, lighting indicators, and controlling relay outputs.

3.4 POWER MANAGEMENT OF SLEEP MODES

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR® provides various sleep modes allowing the user to tailor the power consumption to the application's requirements. When enabled, the brown-out detector (BOD) actively monitors the power supply voltage during the sleep periods. To further save power, it is possible to disable the BOD in some sleep modes. See Section 9.2 "BOD Disable" on page 35 for more details.

3.4.1 SLEEP MODES



	Active Clock Domains					Oscillators		Wake-up Sources							
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other/O	Software BOD Disable
Idle			X	X	X	X	X(2)	X	X	X	X	X	X	X	
ADC noise Reduction				X	X	X	X(2)	X(3)	X	X(2)	X	X	X		
Power-down								X(3)	X				X		X
Power-save					X		X(2)	X(3)	X	X			X		X
Standby(1)						X		X(3)	X				X		X
Extended Standby					X(2)	X	X(2)	X(3)	X	X			X		X

Table 1: sleep modes

To enter any of the six sleep modes, the SE bit in SMCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the SMCR register select which sleep mode (idle, ADC noise reduction, power down, power-save, standby, or extended standby) will be activated by the SLEEP instruction. See Table 9-2 on page 38 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

3.4.2 POWER DOWN MODE

When the SM2...0 bits are written to 010, the SLEEP instruction makes the MCU enter power-down mode. In this mode, the external oscillator is stopped, while the external interrupts, the 2-wire serial interface address watch, and the watchdog continue operating (if enabled). Only an external reset, a watchdog system reset, a watchdog interrupt, a brown-out reset, a 2-wire serial interface address match, an external level interrupt on INT0 or INT1, or a pin change interrupt can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only. Note that if a level triggered interrupt is used for wake-up from power-down mode, the changed level must be held for some time to wake up the MCU. Refer to Section 12. “External Interrupts” on page 53 for details. When waking up from power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped.

3.4.3 POWER SAVE MODE

When the SM2...0 bits are written to 011, the SLEEP instruction makes the MCU enter power-save mode. This mode is identical to power-down, with one exception: If Timer/Counter2 is enabled, it will keep running during sleep. The device can wake up from either timer overflow or output compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK2, and the global interrupt enable bit in SREG is set.

If Timer/Counter2 is not running, power-down mode is recommended instead of power-save mode. The Timer/Counter2 can be clocked both synchronously and asynchronously in power-save mode. If Timer/Counter2 is not using the asynchronous clock, the Timer/Counter oscillator is stopped during sleep. If Timer/Counter2 is not using the synchronous clock, the clock source is stopped during sleep. Note that even if the synchronous clock is running in power-save, this clock is only available for Timer/Counter2.

3.4.4 STAND BY MODE

When the SM2...0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter standby mode. This mode is identical to power-down with the exception that the oscillator is kept running. from standby mode, the device wakes up in six clock cycles

3.5 SYSTEM CONTROL AND RESET

3.5.1 RESET SOURCES

The Atmel ATmega328P has four sources of reset.

Power-on reset. The MCU is reset when the supply voltage is below the power-on reset threshold (VPOT). External reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length. Watchdog system reset. The MCU is reset when the watchdog timer period expires and the watchdog system reset mode is enabled. Brown-out reset. The MCU is reset when the supply voltage VCC is below the brown-out reset threshold (VBOT) and the brown-out detector is enabled

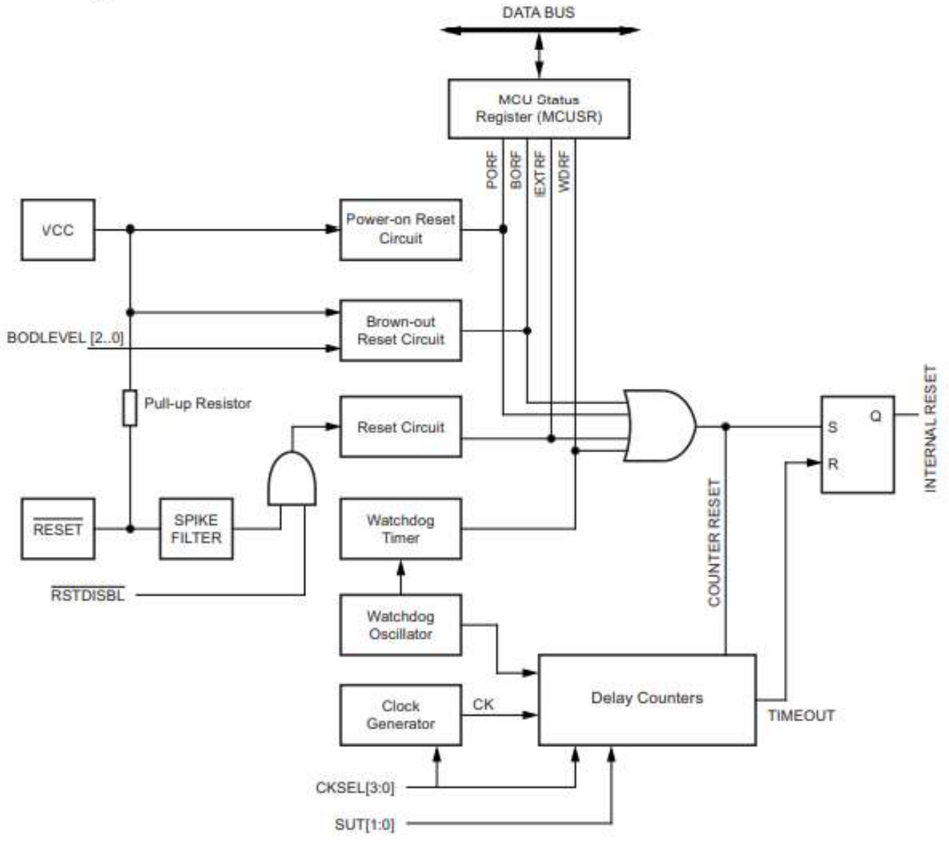


fig 3.3: reset source

3.5.2 POWER ON RESET

A power-on reset (POR) pulse is generated by an on-chip detection circuit. The detection level is defined in Section 28.6 “System and Reset Characteristics” on page 261. The POR is activated whenever VCC is below the detection level. The POR circuit can be used to trigger the start-up reset, as well as to detect a failure in supply voltage. A power-on reset (POR) circuit ensures that the device is reset from power-on. Reaching the power-on reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after VCC rise. The RESET signal is activated again, without any delay, when VCC decreases below the detection level.

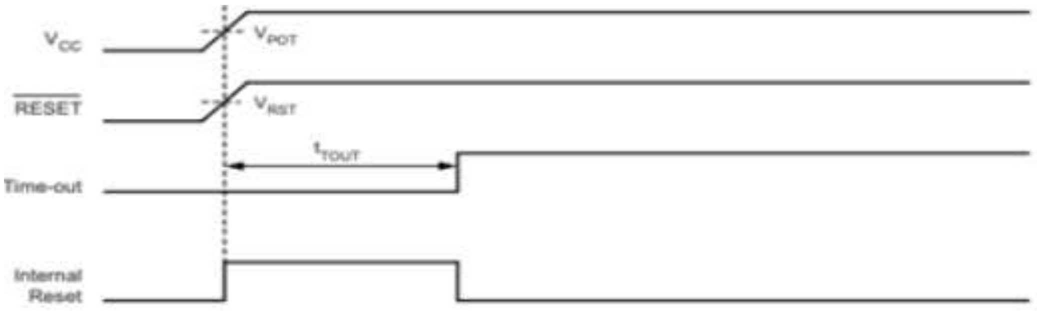


Fig 3.4: MCU start-up, RESET Tired to V_{cc}

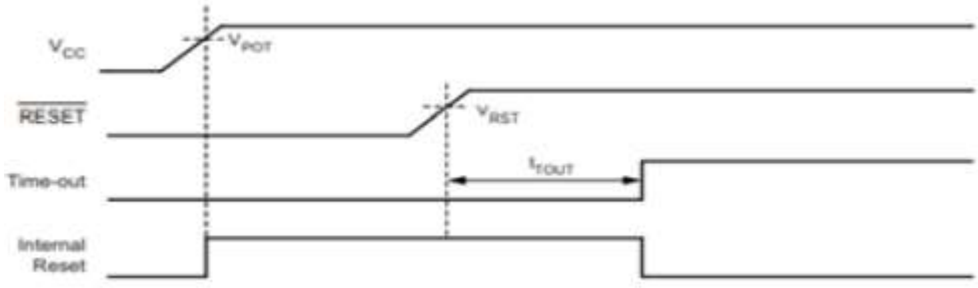
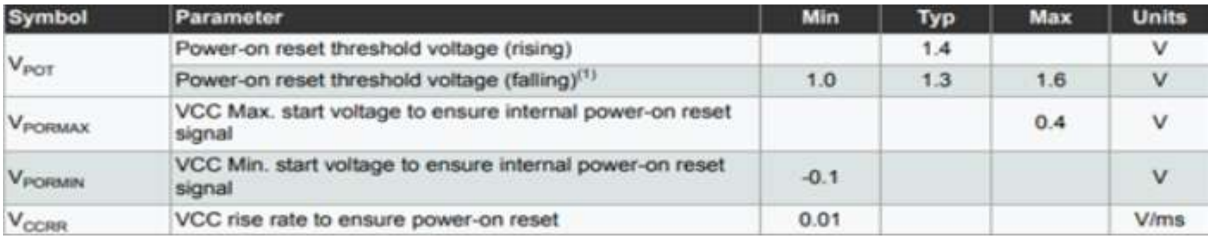


Fig 3.5: MCU start up, RESET extended externally



Symbol	Parameter	Min	Typ	Max	Units
V_{POT}	Power-on reset threshold voltage (rising)		1.4		V
	Power-on reset threshold voltage (falling)(1)	1.0	1.3	1.6	V
V_{PORMAX}	VCC Max. start voltage to ensure internal power-on reset signal			0.4	V
V_{PORMIN}	VCC Min. start voltage to ensure internal power-on reset signal	-0.1			V
V_{CCRR}	VCC rise rate to ensure power-on reset	0.01			V/ms

Table 2: Power on reset specifications

3.5.3 EXTERNAL RESET

An external reset is generated by a low level on the RESET pin. Reset pulses longer than the minimum pulse width (see Section 28.6 “System and Reset Characteristics” on page 261) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the reset threshold voltage – VRST – on its positive edge, the delay counter starts the MCU after the time-out period – tTOUT – has expired. The external reset can be disabled by the RSTDISBL fuse

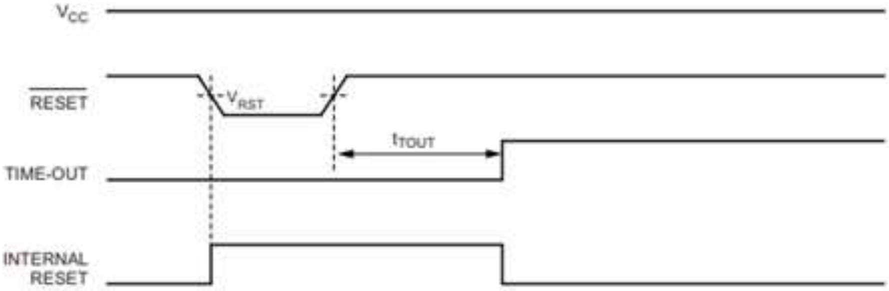


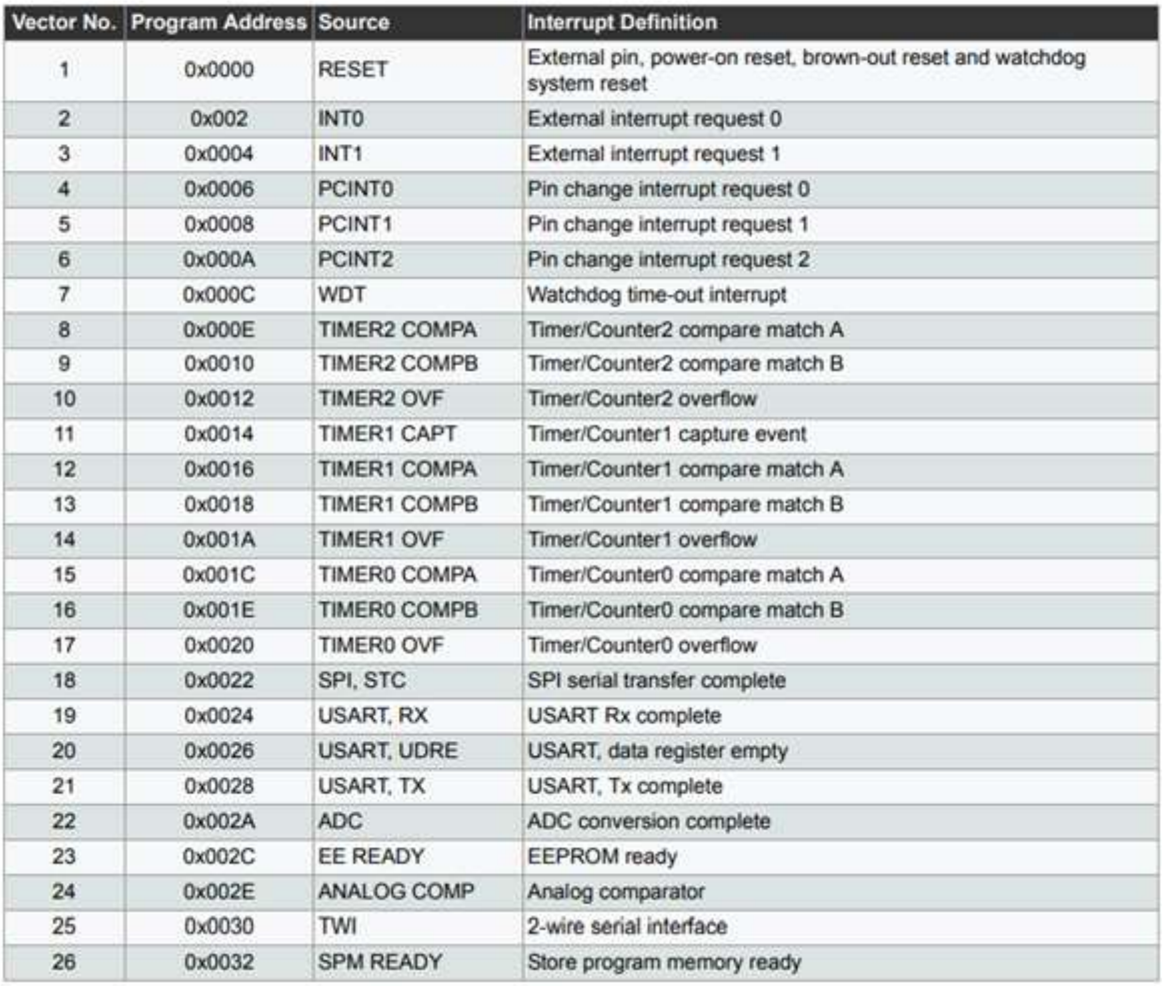
Fig 3.6: external reset during operations

3.6 INTERRUPTS

This section describes the specifics of the interrupt handling as performed in Atmel® Atmega328P. For a general explanation of the AVR® interrupt handling, refer to Section 6.7 “Reset and Interrupt Handling. Each interrupt vector occupies two instruction words in Atmel Atmega328P. In Atmel Atmega328P, the reset vector is affected by the BOOTRST fuse, and the interrupt vector start address is affected by the IVSEL bit in MCUCR.

Another simple **example** of Interrupts is touch screen mobile phones which have the highest priority to the "Touch" sense. Almost every electronic device has some kind to interrupts to 'interrupt' the regular process and do some higher priority things on a particular event. The regular process is resumed after serving the Interrupt.

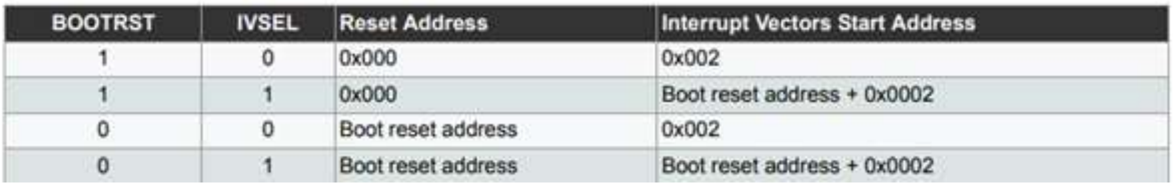
3.6.1 INTERRUPT VECTORS IN AT mega 328P



Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

Table 3: reset and interrupts vectors in ATmega328P

Reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section



BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot reset address + 0x0002
0	0	Boot reset address	0x002
0	1	Boot reset address	Boot reset address + 0x0002

Table 4: reset and interrupt vectors in placement of ATmega328p

3.6.2 EXTERNAL INTERUPTTS

while the interrupt vectors are in the boot section or vice The external interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23...0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23…0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT23...16 pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT14...8 pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT7…0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT23...0 is detected asynchronously.

This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the external interrupt control register A – EICRA. When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in Section 8.1 “Clock Systems and their Distribution” on page 24. Low level interrupt on INT0 and INT1 is detected asynchronously.

The I/O clock is halted in all sleep modes except Idle mode. Note that if a level triggered interrupt is used for wake-up from power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the start-up time, the MCU will still wake up, but no interrupt will be generated.

CHAPTER 4

DESIGNING OF SMART CROP PROTECTION FROM ANIMALS

4.1 INTRODUCTION

In the world, the economy of many countries is dependent upon agriculture. Agriculture is the main stay of economy of the gross domestic product. Agriculture meets food requirements of the people and produces several raw materials for industries. But because of animal interference and fire in agriculture lands, there will be huge loss of crops.

To avoid these financial losses, it is very important to protect agricultural field or farms from animals. Our main purpose of project is to develop intruder alert to the farm, to avoid losses due to animals and fire... This system will not harmful and injurious to animal as well as human beings. The main theme of project is to design an intelligent security system for farm protection by using microcontroller (ARDUINO)

4.2 PROPOSED METHOD OF SMART CROP PROTECTION

- In the proposed system, Crop monitoring is done where sensors are used to collect information in the agricultural field.
- When animals come near to the PIR sensor and it detects the animal movement. After getting that initial input signal, it is passed for further processing. Then it will be given to the microcontroller. Our system will be activated, immediately buzzer will be on and LED glows.
- Microcontroller Block is used for reading the inputs from PIR and Smoke sensor.
- In proposed method of smart crop protection system from animals, Arduino Uno ATmega328p is interfaced with PIR sensor (passive infrared sensor) Which is used to detect the motion of animals or human beings by blinking a led or buzzer.
- The PIR sensor has pyroelectric crystal which detects the heat signatures of living organisms in the fields.
- Existing method of smart crop protection system is Connected as shown in figure

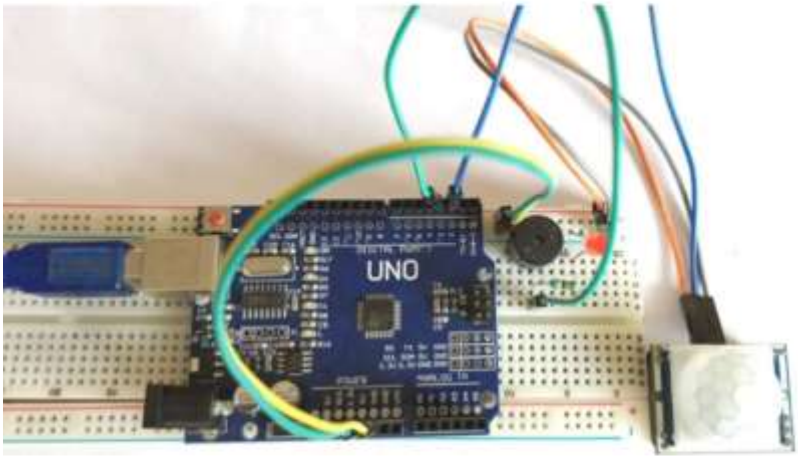


Fig 4.1: proposed method of smart crop protection

4.3 block diagram of smart crop protection system

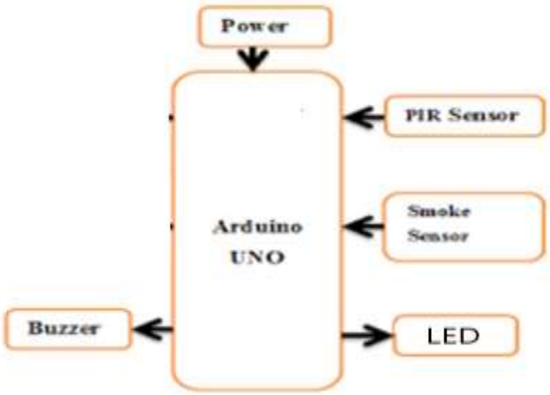


Fig 4.2: Block diagram of smart crop protection system

4.4 TOOLS REQUIRED

4.4.1 SOFTWARE TOOLS

- Arduino IDE 1.8.15(Integrated Development Environment)

The **Arduino Integrated Development Environment** - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

4.4.2 HARDWARE TOOLS

4.4.2.1 ARDUINO UNO ATMEGA328P

Specifications:

Operating voltage	:	5v
Input voltage	:	7v-15v
Analog input pins	:	6(A0-A5)

➢ Digital I/O pins	:	14(D0-D13)
➢ Flash memory	:	32kb
➢ Frequency	:	16MHz
➢ EEPROM	:	1KB
➢ SRAM	:	2KB

4.4.2.2 PIR SENSOR

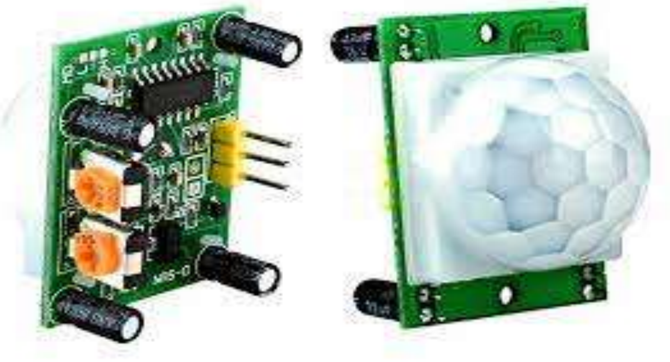


Fig 4.3: PIR sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. PIRs are basically made of a pyroelectric sensor, which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Specifications:

- It has three input pins.

  1)Ground

  2)VCC

  3)Signal
- Operating voltage : 5v
- Actual detection range : 5m-12m
- Frequency : 60MHz
- Power supply : 5V-12V
- Regulated voltage : 3.3V

4.4.2.3 BUZZER

Fig 4.4: buzzer

A buzzer is a loud noise maker. Most modern ones are civil defense or air-raid sirens, tornado sirens, or the sirens on emergency service vehicles such as ambulances, police cars and fire trucks. There are two general types, pneumatic and electronic

Specifications :

- Rated Voltage : 6V DC
- Operating Voltage : 4 to 8V DC
- Rated Current : ≤30mA
- Sound Output at 10cm : ≥85dB

- Resonant Frequency : 2300 ±300Hz

4.4.2.3 LED

Fig 4.5 : LED

Made popular by their efficiency, range of color, and long lifespan, LED lights are ideal for numerous applications including night lighting, art lighting, and outdoor lighting. These lights are also commonly used in electronics and automotive industries, and for signage, along with many other uses.

4.5 CIRCUIT DIAGRAM

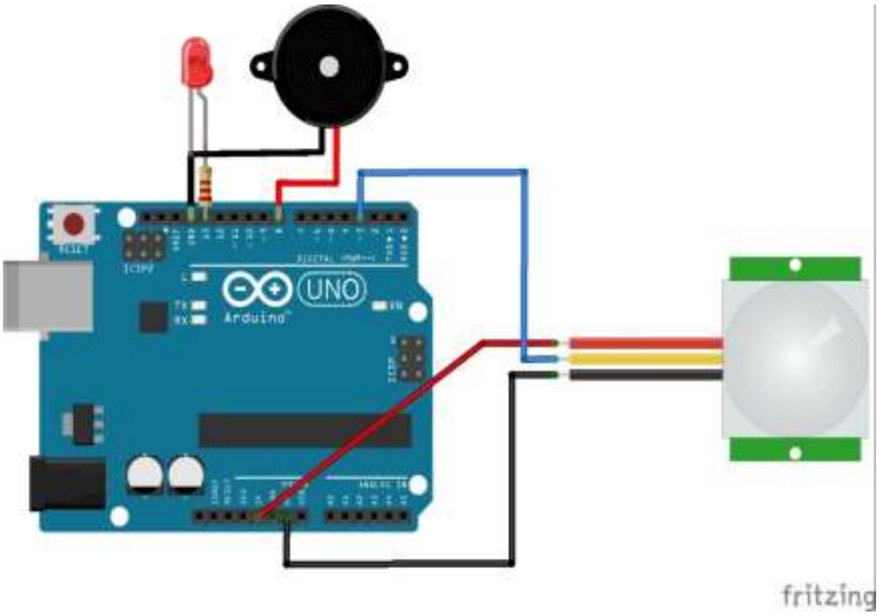


Fig 4.6: circuit diagram of smart crop protection system from animals

1). PIR to Arduino

- Connect the V_{cc} of PIR to 5V on Arduino

- Connect the GND of PIR to GND on Arduino
- Connect the OUTPUT pin of PIR to Digital pin D3 on Arduino

2). Buzzer to Arduino

- Connect one pin of buzzer to digital pin D8 on Arduino
- Connect other pin of buzzer to GND on Arduino

3). LED to Arduino

- Connect the LED positive to Digital pin D13 on Arduino through a resistor.
- Connect the LED negative to GND on Arduino.

Working

PIR sensor is a special type sensor which is usually used for security purposes. It detects the objects by reading the Infrared radiations emitted by the objects. Any object whose temperature is above absolute zero, emits radiation. This radiation is not visible to human eyes. The PIR sensor is designed to detect this Infrared radiation.

The PIR sensor has two modes. You can switch between these modes by interchanging the jumper behind the PIR sensor as shown in the images below...

a). Single trigger mode

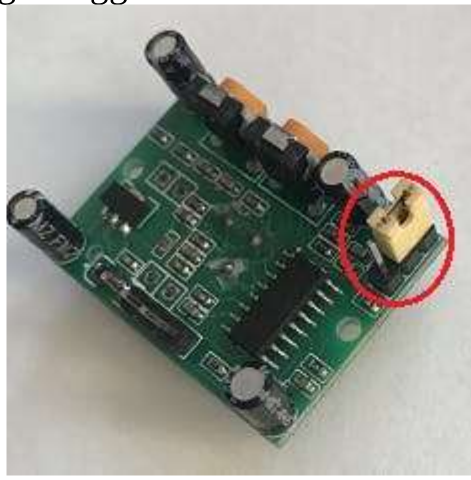


Fig 4.7: single trigger mode

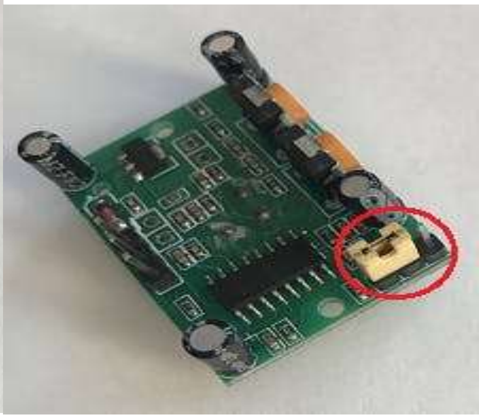


Fig4.8: Repeatable trigger mode

4.6 RESULT

4.6.1 LED ON

When PIR sensors DETECTS the motion of the animals and that detected signal is transmitted to Arduino UNO in that case LED is ENABLE. If PIR sensor does not detects any motion the LED is DISABLE

4.6.2 BUZZER ON

When PIR sensors DETECTS the motion of the animals and that detects signals send to the Arduino and it gives the alert signal to buzzer that case BUZZER ON. In case does not detects any motion BUZZER OFF state.

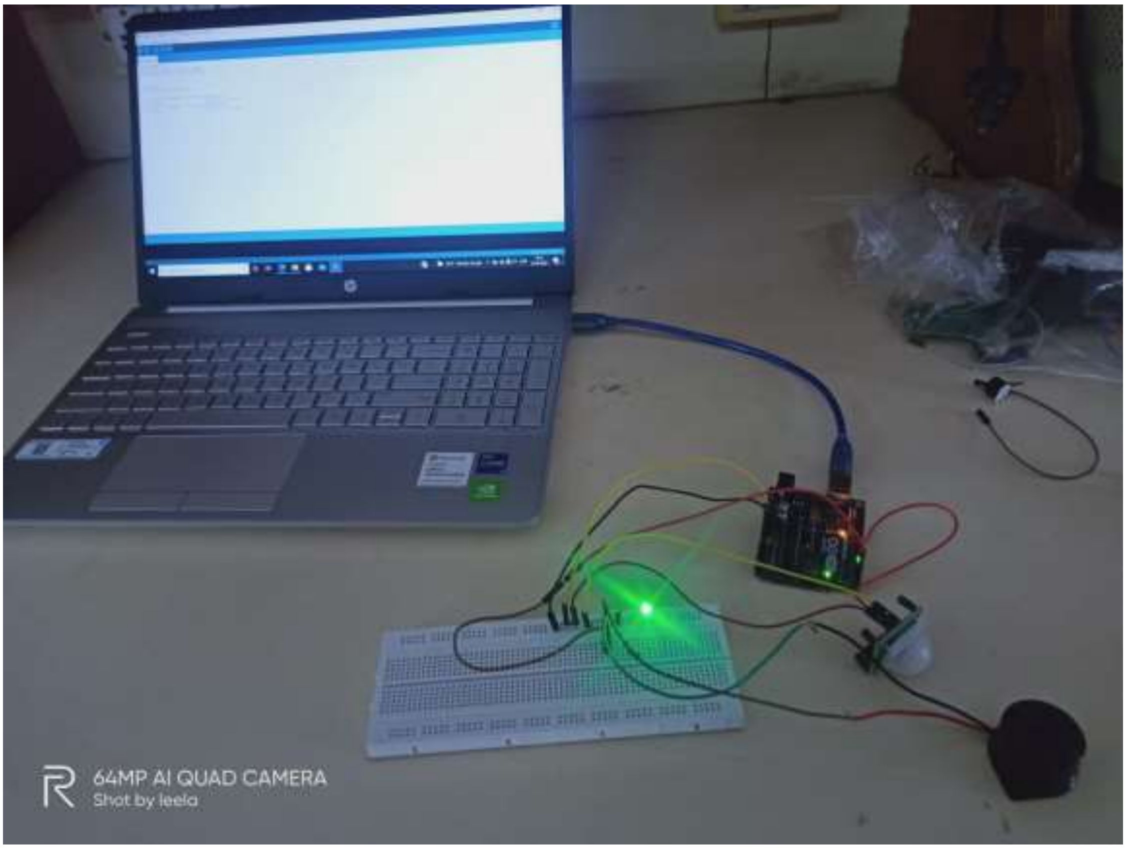


Fig 4.6 : output of smart crop protection system from animals

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The problem of crop vandalization by wild animals and fire has become a major social problem in current time. It requires urgent attention as no effective solution exists till date for this problem. Thus, this project carries a great social relevance as it aims to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic wellbeing.

5.2 FUTURE SCOPE

In the future, there will be very large scope, this project can be made based on Image processing in which wild animal and fire can be detected by cameras and if it comes towards farm then system will be directly activated through wireless networks. Wild animals can also be detected by using wireless networks such as laser wireless sensors and by sensing this laser or sensor’s security system will be activated and also GSM module also, we can try.

REFERENCE

[1] Dr. Wilson, "ELECTRIC FENCE," Handbook of Texas, Project report published by the Texas State Historical Association. August 4, 2011

[2] T. Day and R. Mac Gibbon, “Multiple-Species Exclusion Fencing and Technology for Mainland Sites.”, Project Report published by National Wildlife Research Centre, 2007

[3] R. Padula and W. Head, “Fencing System” Project Report published by University of Minnesota, 2003

[4] Astif Bherani, Gaurav Kumar N. raut, Pawan D. kale “smart design of microcontroller monitoring system for agriculture,” international conference on circuit, power and computing technologies, IEEE 2014.

[5] Wang, Z., Wang, H., Liu, L., Song, W., & Lu, J. (2015, December). Community alarm system design based on MCU and GSM. In 2015 4th International Conference on Computer Science and Network Technology (ICCSNT) (Vol. 1, pp. 859-862). IEEE.

[6] Shende, P. Y., Raut, S. M., Ingale, P. S., Nagose, A. K., Katakpur, P. S., & Kathane, S. S. Solar Electric Fencing for Irrigation of Animal Man Conflict

[7] Mohammad, T. (2009). Using ultrasonic and infrared sensors for distance measurement. World Academy of Science, Engineering and Technology, 51, 293-299

[8] Volume:01 Pages:859-862, DOI:10.1109/ICCSNT.2015.7490876, IEEE Conference Publications

[9] Amin, N.& Borschbach, M. (2015, December). Quality of obstacle distance measurement using ultrasonic sensor and precision of two computer vision-based obstacle detection approaches. In 2015 International Conference ion Smart Sensors and Systems i(IC-SSS) i(pp. i1-6). IEEE.

[10] Pages:1-6, DOI: 10.1109/SMARTSENS.2015.7873595IEEE Conference Publications

[11] Mustapha, B. Zayegh, A., & Begg, R. K. (2013, December). Ultrasonic and infrared sensors performance in a wireless obstacle

detection system. In 2013 1st International Conference on Artificial Intelligence, Modelling and Simulation (pp. 487-492). IEEE.

[12] DOI: 10.1109/IECON.2013.6700389 CITED BY: PAPERS (1)

[13] Frankiewicz, A., & Cupek, R. (2013, November). Smart passive infrared sensor-Hardware platform. In IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society (pp. i7543-7547). IEEE.

[14] Discant, A. Rogozan, C. Rusu and A. Bensrhair, “Sensors for Obstacle Detection” 2007 30th International Spring Seminar on Electronics Technology (ISSE), Cluj-Napoca, 2007, pp.100-105. do: i10.1109/ISSE.2007.44328