CHAPTER 1: INTRODUCTION

i. Graduation Project:-

Is a robot security project "Security Guard intelligent" works in a company.

This depends on the basis of the guard plate Arduino is used for security purposes.

ii. Introduction to the guard

The security guard is a guard robot that has the company and contains the alarms and sensors such as temperature sensor and a sound sensor and a motion sensor and a smoke sensor.

iii. Components to the guard

- ✓ **Arduino Mega 2560**
- ✓ **9V Battery Adapter for the Arduino**
- ✓ **High Quality Arduino USB programming Cable**
- ✓ **Sound Sensor (adjustable)**
- ✓ **PIR Motion sensor module (Adjustable Range)**
- ✓ **LM35DZ Temperature Sensor (Precision Centigrade)**
- ✓ **Smoke sensor Module MQ2 (Digital/Analog)**

iv. This is the modus operandi of the guard

This guard works as a security guard in a company.

If this fire happened senses goalkeeper fire by heat sensor is called the voice of alarm through the sensor sound that exists inside the goalkeeper .ki notice of workers and employees in this company to the fire.

In order to deal with the fire firefighters cautiously.

If there is smoke senses this guard by the smoke sensor smoke and called the voice of alarm through the sensor sound that exists inside the goalkeeper .ki notice of workers and employees in this company to smoke.

In order to deal with the smoke firefighters cautiously.

There are in this goalkeeper motion sensor so that the guard is moving this device to all places of the company located out

1.1. Introduction to Microcontroller

1.1.1. Introduction

Microcontrollers have only been with us for a few decades but their impact (direct or indirect) on our lives is profound. Usually these are supposed to be just data processors performing exhaustive numeric operations.

But their presence is unnoticed at most of the places like

- **At supermarkets in Cash Registers, Weighing Scales, etc.**
- **At home in Ovens, Washing Machines, Alarm Clocks, etc.**
- **At play in Toys, VCRs, Stereo Equipment, etc.**
- **At office in Typewriters, Photocopiers, Elevators, etc.**
- **In industry in Industrial Automation, safety systems, etc.**
- **On roads in Cars, Traffic Signals, etc.**

What inside them makes these machines smart?

The answer is microcontroller:

Creating applications for the microcontrollers is different than any other development job in electronics and computing. Before selecting a particular device for an application, it is important to understand what the different options and features are and what they can mean with regard to developing the application.

The purpose of this chapter is to introduce the concept of a microcontrollers, how it different from microprocessors, different type of commercial microcontrollers available as well as their applications. The reminder of the book will go through and present different types of microcontrollers and also programming and interfacing techniques of microcontroller, mainly 8051, in detail.

1.1.2. Embedded Controller

Simply an embedded controller is a controller that is embedded in a greater system. One can define an embedded controller as a controller (or computer) that is embedded into some device for some purpose other than to provide general purpose computing.

Is an embedded controller is the same as a microcontroller? The answer is definitely no. One can state devices such as 68000, 32032, x86, Z80, and so on that are used as embedded controllers but they aren't micro-controllers

We might be correct by stating that an embedded controller controls something (for example controlling a device such as a microwave oven, car braking system or a cruise missile). An embedded controller may also embed the on-chip resources like a microcontroller. Microcontrollers and microprocessors are widely used in embedded systems. Though microcontrollers are preferred over microprocessors for embedded systems due to low power consumption.

1.1.3. Microcontrollers and Microprocessors

A controller is used to control some process. At one time, controllers were built exclusively from logic components, and were usually large, heavy boxes. Later on, microprocessors were used and the entire controller could fit on a small circuit board. This is still commonone can find many controllers powered by one of the many common microprocessors (including Zilog Z80, Intel 8088, Motorola 6809, and others).

As the process of miniaturization continued, all of the components needed for a controller
were built right onto one chip. A one chip computer, or microcontroller was born.

A CPU built into a single VLSI chip is called microprocessor. The simplified block diagram of the CPU is shown in the Fig. 1.1. It contains arithmetic and logic unit (ALU), Instruction decode and control unit, Instruction register, Program counter (PC), clock circuit (internal or external), reset circuit (internal or external) and registers. For example, Intel 8085 is 8-bit microprocessor and Intel 8086/8088 is 16-bit microprocessor.

Microprocessor is general-purpose digital computer central processing unit (CPU). The microprocessor is general-purpose device and additional external circuitry are added to make it microcomputer.

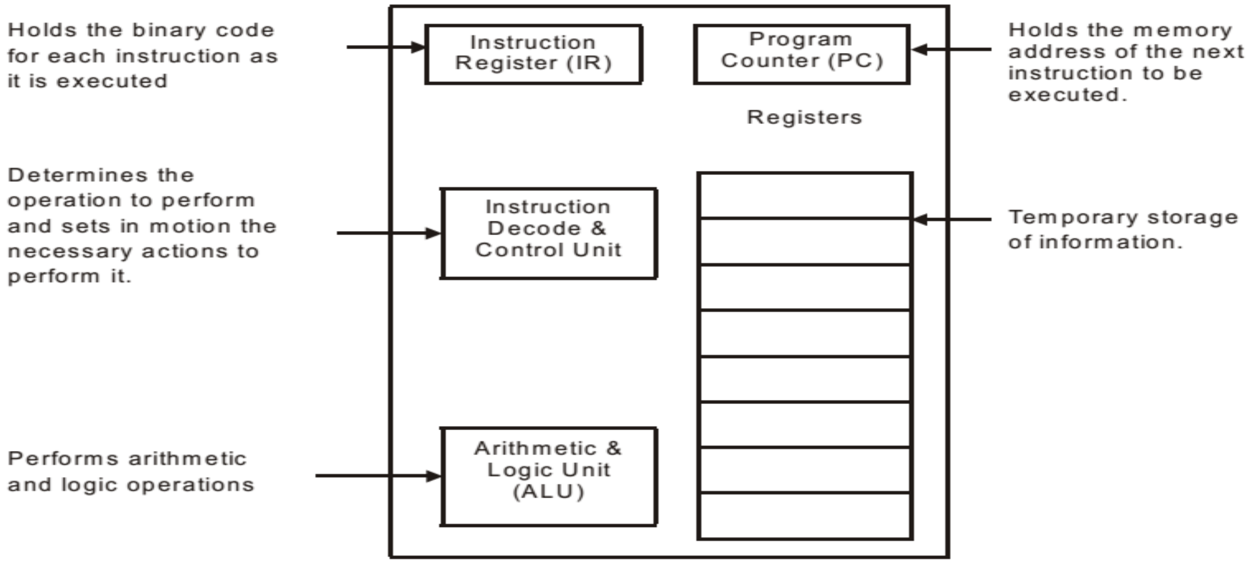


Figure 1.1 General block diagram of CPU (Microprocessor)

A digital computer having microprocessor as the CPU along with I/O devices and memory is known as microcomputer. The block diagram in the Fig. 1.2 shows a microcomputer.

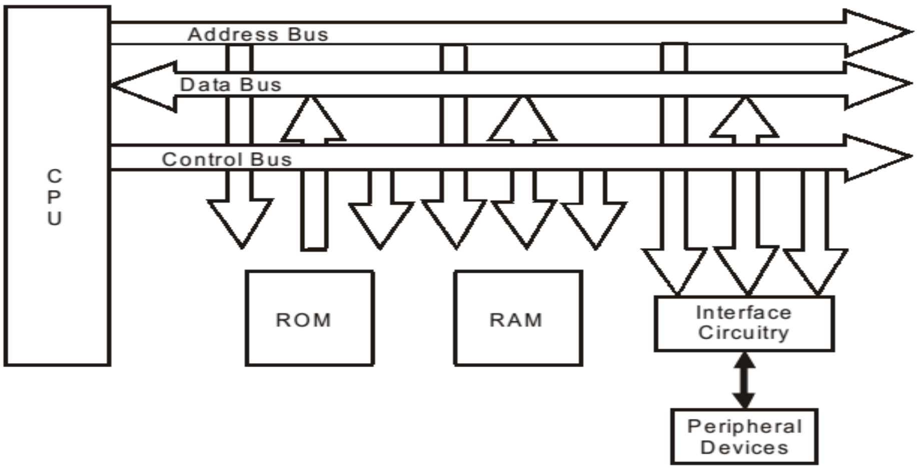


Figure 1.2 Microcomputer block diagram

A microcontroller is a highly integrated chip, which includes on single chip, all or most of the parts needed for a controller. The microcontroller typically includes: CPU (Central Processing Unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output) - serial and parallel, timers, interrupt controller.

For example, Intel 8051 is 8-bit microcontroller and Intel 8096 is 16-bit microcontroller.

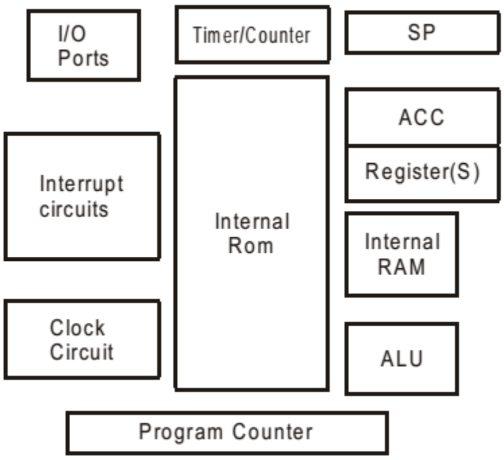


Figure 1.3 A block diagram of a microcontroller

By only including the features specific to the task (control), cost is relatively low. A typical microcontroller has bit manipulation instructions, easy and direct access to I/O (input/output), and quick and efficient interrupt processing. Figure 1.3 shows the block diagram of a typical microcontroller.

COMPARING MICROPROCESSORS AND MICROCONTROLLERS :

- **Microprocessor is a single chip CPU, microcontroller contains, a CPU and much of the remaining circuitry of a complete microcomputer system in a single chip.**
- **Microcontroller includes RAM, ROM, serial and parallel interface, timer, interrupt**

  schedule circuitry (in addition to CPU) in a single chip.
  - **RAM is smaller than that of even an ordinary microcomputer, but enough for its applications.**
  - **Interrupt system is an important feature, as microcontrollers have to respond to control oriented devices in real time. E.g., opening of microwave oven's door cause an interrupt to stop the operation. (Most microprocessors can also implement powerful interrupt schemes, but external components are usually needed).**
- **Microprocessors are most commonly used as the CPU in microcomputer systems.**
- **Microcontrollers are used in small, minimum component designs performing control-oriented activities.**

- **Microprocessor instruction sets are processing intensive , implying powerful addressing modes with instructions catering to large volumes of data. Their instructions operate on nibbles, bytes, etc.**
- **Microcontrollers have instruction sets catering to the control of inputs and outputs. Their instructions operate also on a single bit. E.g., a motor may be turned ON and OFF by a 1-bit output port.**

Before going in to details of microcontrollers it will be beneficial to go through common and frequently used terminology encountered in the description of microcontrollers.

CENTRAL PROCESSING UNIT (CPU) :

CPU is the brain of the computer system, administers all activity in the system and performs all operations on data. It continuously performs two operations: fetching and executing instructions. It understand and execute instructions based on a set of binary codes called the instruction set.

Machine Cycle

To execute an instruction-the processor must:

1. **Fetch the instruction from memory**
2. **Decode the instruction**
3. **Execute the instruction**
4. **Store the result back in the memory. These four steps refer to Machine Cycle.**

Generally one machine cycle = X clock cycles(X depends on the particular instruction being executed). Shorter the clock cycle, lesser the time it takes to complete one machine cycle, so instructions are executed faster. Hence, faster the processor.

FETCHING AND EXECUTING AN INSTRUCTION

Fetching involves the following steps:

(a) Contents of PC are placed on address bus.

(b) READ signal is activated.

(c) Data (instruction opcode) are read from RAM and placed on data bus.

(d) Opcode is latched into the CPU's internal instruction register.

(e) PC is incremented to prepare for the next fetch from memory.

While execution involves decoding the opcode and generating control signals to gate internal registers in and out of the ALU and to signal the ALU to perform the specified operation.

THE BUSES: ADDRESS, DATA, AND CONTROL

A BUS is a collection of wires carrying information with a common purpose. For each read or write operation, the CPU specifies the location of the data or instruction by placing an address on the address bus, then activates a signal on the control bus indicating whether the operation is read or write.

**-READ OPERATIONS retrieve a byte of data from memory at the location specified
and place it on the data bus. CPU reads the data and places it in one of its internal
registers.**

-WRITE OPERATIONS put data from CPU on the data bus and store it in the location specified.

ADDRESS BUS carries the address of a specified location. For n address lines, 2^n locations can be accessed. E.g., A 16-bit address bus can access 2^{16} = 65,536 locations or 64K locations (2^{10} = 1024 = 1K,2^6 = 64).

DATA BUS carries information between the CPU and memory or between the CPU and I/O devices.

CONTROL BUS carries control signals supplied by the CPU to synchronize the movement of information on the address and data bus.

CONTROL/MONITOR (INPUT/OUTPUT) DEVICES

CONTROL DEVICES are outputs, or actuators, that can affect the world around them when supplied with a voltage or current.

**MONITORING DEVICES are inputs, or sensors, that are stimulated by temperature,
pressure, light, motion, etc. and convert this to voltage or current read by the computer.**

Note: The interface circuitry converts the voltage or current to binary data, or vice versa.

1.1.4. TYPES OF MICROCONTROLLERS

Microcontrollers can be classified on the basis of internal bus width, architecture, memory and instruction set. Figure 1.4 shows the various types of microcontrollers.

1.1.4.1 THE 8, 16 AND 32-BIT MICROCONTROLLERS

THE 8-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a byte (8-bits) at an instruction, the microcontroller is an 8-bit microcontroller. The internal bus width of 8-bit microcontroller is of 8-bit. Examples of 8-bit microcontrollers are Intel 8051 family and Motorola MC68HC11 family.

THE 16-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a word (16-bits) at an instruction, the microcontroller is an 16-bit microcontroller. The internal bus width of 16-bit microcontroller is of 16-bit. Examples of 16-bit microcontrollers are Intel 8096 family and Motorola MC68HC12 and MC68332 families. The performance and computing capability of 16 bit microcontrollers are enhanced with greater precision as compared to the 8-bit microcontrollers.

THE 32-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a double word (32-bits) at an instruction, the microcontroller is an 32-bit microcontroller. The internal bus width of 32-bit microcontroller is of 32-bit. Examples of 32-bit microcontrollers are Intel 80960 family and Motorola M683xx and Intel/Atmel 251 family. The performance and computing capability of 32 bit microcontrollers are enhanced with greater precision as compared to the 16-bit microcontrollers.

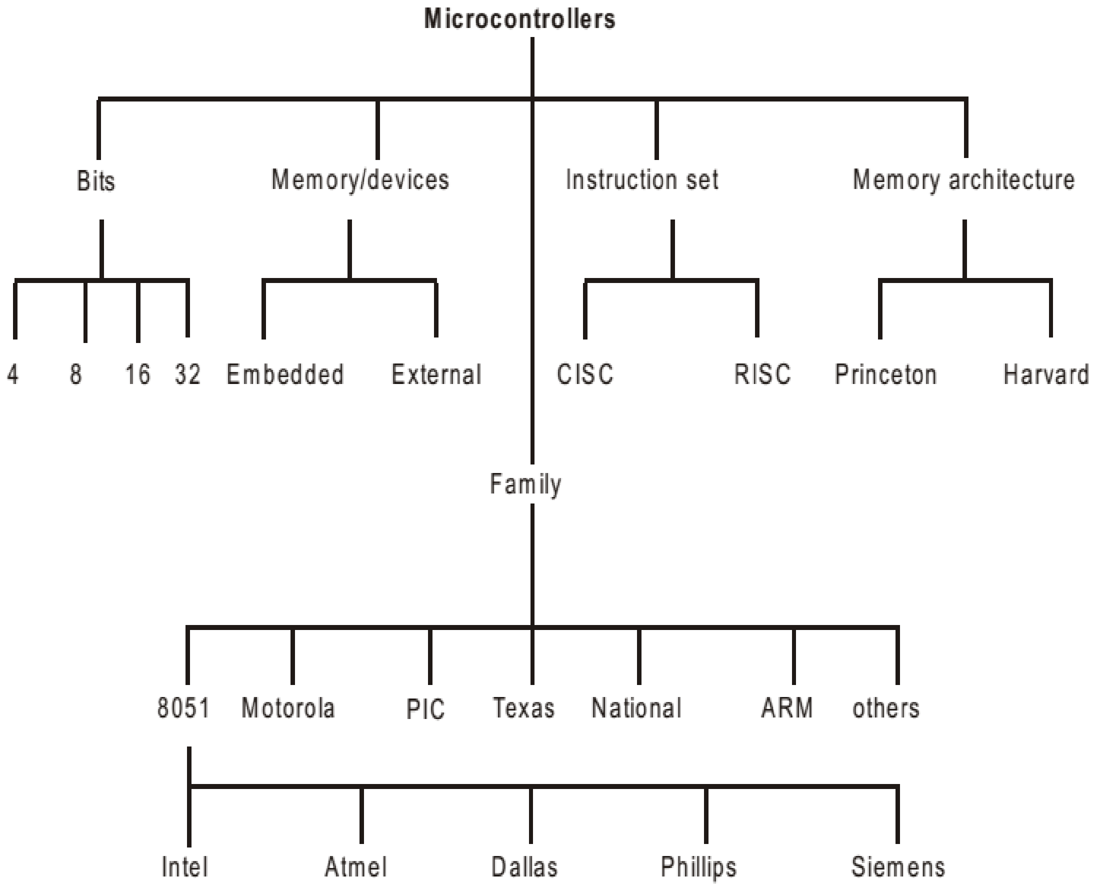


Figure 1.4 Types of microcontrollers

1.1.4.2 EMBEDDED AND EXTERNAL MEMORY MICROCONTROLLERS

EMBEDDED MICROCONTROLLERS

When an embedded system has an microcontroller unit that has all the functional blocks (including program as well as data memory) available on a chip is called an embedded microcontroller. For example, 8051 having Program & Data Memory, I/O Ports, Serial Communication, Counters and Timers and Interrupt Control logic on the chip is an embedded microcontroller.

EXTERNAL MEMORY MICROCONTROLLERS

When an embedded system has an microcontroller unit that has not all the functional blocks available on a chip is called an external memory microcontroller. In external memory microcontroller, all or part of the memory units are externally interfaced using an interfacing circuit called the glue circuit. For example, 8031 has no program memory on the chip is an external memory microcontroller.

1.1.4.3 MICROCONTROLLER ARCHITECTURAL FEATURES

There are mainly two categories of processors, namely, Von-Neuman (or Princeton) architecture and Harvard Architecture. These two architecture differ in the way data and programs are stored and accessed.

1.1.4.3.1 VON-NEUMAN ARCHITECTURE

Microcontrollers based on the Von-Neuman architecture have a single data bus that is used to fetch both instructions and data. Program instructions and data are stored in a common main memory. When such a controller addresses main memory, it first fetches an instruction, and then it fetches the data to support the instruction.
The two separate fetches slows up the controller's operation. Figure 1.5 shows the Von-Neuman Architecture. The Von-Neuman architecture's main advantage is that it simplifies the microcontroller design because only one memory is accessed.

In microcontrollers, the contents of RAM can be used for data storage and program instruction storage. For example, the Motorola 68HC11 microcontroller Von-Neuman architecture.

Example : An Instruction "Read a byte from memory and store it in the accumulator" as follows:

Cycle 1 :- Read instrution

Cycle 2 - Read data out of RAM and put into Accumulator

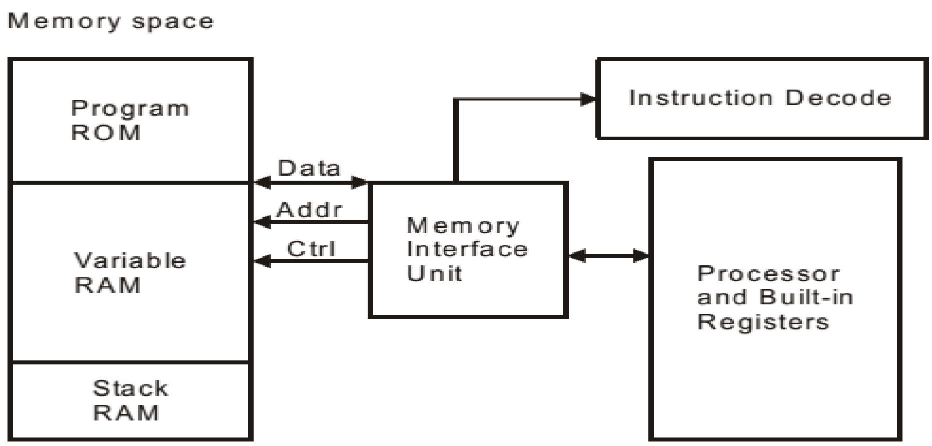


Figure 1.5 Von-neuman architecture block diagram

1.1.4.3.2 HARVARD ARCHITECTURE

Microcontrollers based on the Harvard Architecture have separate data bus

and an instruction bus. This allows execution to occur in parallel. As an instruction is being "pre-fetched", the current instruction is executing on the data bus. Once the current instruction is complete, the next instruction is ready to go. This pre-fetch theoretically allows for much faster execution than Von-Neuman architecture, on the expense of complexity. Figure 1.6 shows the Harvard Architecture.The Harvard Architecture executes instructions in fewer instruction cycles than the Von-Neuman architecture.

For example,the intel MCS-51 family of microcontrollers and PIC microcontrollers uses Harvard Architecture.

The same instruction (as shown under Von-Newman architecture) would be executed as follows:

Cycle 1: - Complete previous instructio

- Read the "Move Data to Accumulator" instruction

Cycle 2: - Execute "Move Data to Accumulator" instruction

- Read next instruction

Hence each instruction is effectively executed in one instruction cycle.

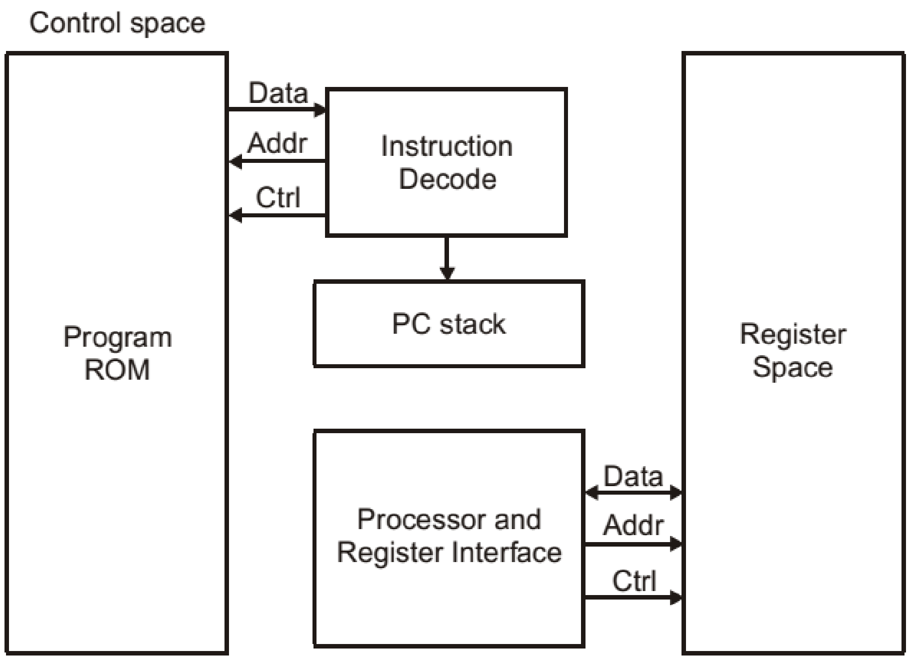


Figure 1.6 Harvard architecture block diagram

1.1.4.3.3 CISC (COMPLEX INSTRUCTION SET COMPUTER)

ARCHITECTURE

MICROCONTROLLERS

Almost all of today's microcontrollers are based on the CISC (Complex Instruction Set Computer) concept. When an microcontroller has an instruction set that supports many addressing modes for the arithmetic and logical instructions, data transfer and memory accesses instructions, the microcontroller is said to be of CISC architecture.

The typical CISC microcontroller has well over 80 instructions, many of them very

powerful and very specialized for specific control tasks. It is quite common for the instructions to all behave quite differently. Some might only operate on certain address spaces or registers, and others might only recognize certain addressing modes.

The advantages of the CISC architecture are that many of the instructions are macrolike, allowing the programmer to use one instruction in place of many simpler instructions. An example of CISC architecture microcontroller is Intel 8096 family.

1.1.4.3.4 RISC (REDUCED INSTRUCTION SET COMPUTER) ARCHITECTURE

MICROCONTROLLERS

The industry trend for microprocessor design is for Reduced Instruction Set Computers (RISC) designs. When a microcontroller has an instruction set that supports fewer addressing modes for the arithmetic and logical instructions and for data transfer instructions, the microcontroller is said to be of RISC architecture.

The benefits of RISC design simplicity are a smaller chip, smaller pin count, and very low power consumption.

Some of the typical features of a RISC processor- Harvard architecture are

1. **Allows simultaneous access of program and data.**
2. **Overlapping of some operations for increased processing performance.**
3. **Instruction pipelining increases execution speed.**
4. **Orthogonal (symmetrical) instruction set for programming simplicity.**
5. **Allows each instruction to operate on any register or use any addressing mode.**

1.1.4.3.5 SISC (SPECIFIC INSTRUCTION SET COMPUTER)

Actually, a microcontroller is by definition a Reduced Instruction Set Computer. It could really be called a Specific Instruction Set Computer (SISC). The basic idea behind the microcontroller was to limit the capabilities of the CPU itself, allowing a complete computer (memory, I/O, interrupts, etc) to fit on the single chip. At the expense of the more general purpose instructions that make the standard microprocessors (8088, 68000, 32032) so easy to use, the instruction set was designed for the specific purpose of control (powerful bit manipulation, easy and efficient I/O, and so on).

1.1.5. Microcontroller Applications

In addition to control applications such as the home monitoring system, microcontrollers are frequently found in embedded applications. Among the many uses that you can find one or more microcontrollers: automotive applications, appliances (microwave oven, refrigerators, television and VCRs, stereos), automobiles (engine control, diagnostics, climate control), environmental control (greenhouse, factory, home), instrumentation, aerospace, and thousands of other uses.

Microcontrollers are used extensively in robotics.

In this application, many specific tasks might be distributed among a large number of microcontrollers in one system.

Communications between each microcontroller and a central, more powerful microcontroller (or microcomputer, or even large computer) would enable information to be processed by the central computer, or to be passed around to other microcontrollers in the system.

A special application that microcontrollers are well suited for is data logging.

By stick one of these chips out in the middle of a corn field or up in a balloon, one can monitor and record environmental parameters (temperature, humidity, rain, etc). Small size, low power consumption, and flexibility make these devices ideal for unattended data monitoring and recording.

1.1.6. COMMERCIAL MICROCONTROLLER DEVICES

Microcontrollers come in many varieties. Depending on the power and features that are needed, one might choose a 4 bit, 8 bit, 16 bit, or 32 bit microcontroller.

In addition, some specialized versions are available which include features specific for communications, keyboard handling, signal processing, video processing, and other tasks.

The examples of different types of commercial microcontroller devices are given in the following tables.

Table 1.1 4 Bit Microcontrollers

Extra Features	Counters	ROM (bytes)	RAM (bytes)	Pins	I/O	Model (Manufacturer)
Serial bit I/O	1	1K	64	28	23	COP400 Family (National)
10-bit ROM	-	512	32	28	10	HMCS40 (Hitachi)
LED display	-	1K	64	28	23	TMS 1000 (Texas Instruments)

Table 1.2 8 Bit Microcontrollers

Extra Features	Counters	ROM (bytes)	RAM (bytes)	Pins	I/O	Model (Manufacturer)
8k External memory	1	1K	64	40	27	8048 (Intel)
128k External memory, Boolean processing, serial port	2	4K	128	40	32	8051 (Intel)
Serial bit I/O, 8- channel A/D converter	1	1K	64	28	24	COP800 Family
PLL frequency synthesizer, A/D, PWM generator,	1 2	1K 8K	64 256	28 52	20 40	6805 (Motorola) 68hc11(Motorola)
watchdog timer, Instru- ments) Serial ports,	2	4K	256	68	55	TMS370 (Texas)
small pin count, very low power consumption	0	1K	25	18	12	PIC (Micro Chip)

Table 1.3. 16 Bit Microcontrollers

Extra Features	Counters	ROM (bytes)	RAM (bytes)	Pins	I/O	Model (Manufacturer)
PWM generator, watchdog		2	232 8K	68	40	80c196 (Intel)
PWM generator, watchdog timer, 8-channel		4	512 16K	68	52	HPC Family (National)

1.2 Intoduction to Arduino

1.2.1 Introduction

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures kits for building digital devices and interactive objects that can sense and control the physical world. Arduino boards may be purchased preassembled, or as do-it-yourself kits; at the same time, the hardware design information is available for those who would like to assemble an Arduino from scratch.

The project is based on a family of microcontroller board designs manufactured primarily by SmartProjects in Italy,and also by several other vendors, using various 8-bit Atmel AVR microcontrollers or 32-bit Atmel ARM processors. These systems provide sets of digital and analog I/O pins that can be interfaced to various extension boards and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for C and C++ programming languages.

The first Arduino was introduced in 2005. The project leaders sought to provide an inexpensive and easy way for hobbyists, students, and professionals to create devices that interact with their environment using sensors and actuators. Common examples for beginner hobbyists include simple robots, thermostats and motion detectors. Adafruit Industries estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands

1.2.2 History

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of $100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea. The name "Arduino" comes from a bar in Ivrea, where some of the founders of the project used to meet. The bar itself was named after Arduino, Margrave of Ivrea and King of Italy from 1002 to 1014.

A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the Wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community. The school eventually closed, but the researchers, including David Cuartielles, promoted the idea.

1.2.3 Hardware

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as *shields*. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus—so many shields can be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, 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 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 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

There are many Arduino-compatible and Arduino-derived boards. 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 to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor—sometimes retaining compatibility with shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

1.2.4 Official boards

The original Arduino hardware is manufactured by the Italian company Smart Projects. Some Arduino-branded boards have been designed by the American company SparkFun Electronics. Sixteen versions of the Arduino hardware have been commercially produced to date

1.2.5 Software

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a *sketch*.

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

A typical first program for a microcontroller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this:

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a *sketch*.

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

A typical first program for a microcontroller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this:

```
#define LED_PIN 13

void setup () {
  pinMode (LED_PIN, OUTPUT); // Enable pin 13 for digital output
}

void loop () {
  digitalWrite (LED_PIN, HIGH); // Turn on the LED
  delay (1000); // Wait one second (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // Turn off the LED
```

```
  delay (1000); // Wait one second
}
```

It is a feature of most Arduino boards that they have an LED and load resistor connected between pin 13 and ground; a convenient feature for many simple tests. The previous code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board" button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program.

The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.

As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

1.2.6 Development

Arduino is open source hardware: the Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE is available and released under the GNU General Public License, version 2.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derivative works without permission. The official policy document on the 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 "Arduino" name by using "-duino" name variants.

1.2.7 Applications

See also: List of open source hardware projects

- **Xoscillo: open-source oscilloscope**
- **Scientific equipment**
- **Arduinome: a MIDI controller device that mimics the Monome**
- **OBDuino: a trip computer that uses the on-board diagnostics interface found in most modern cars**
- **Ardupilot: drone software / hardware**
- **ArduinoPhone**
- ***GertDuino*, an Arduino mate for the Raspberry Pi**
- **Water quality testing platform**

1.3 Reception

The Arduino project received an honorary mention in the Digital Communities category at the 2006 Prix Ars Electronica.

1.4 See also

- **List of Arduino boards and compatible systems**
- **Comparison of single-board computers**

1.2.8 Ultrasonic transducer

Ultrasonic transducers are transducers that convert ultrasound waves to electrical signals or vice versa. Those that both transmit and receive may also be called ultrasound transceivers; many ultrasound sensors besides being sensors are indeed transceivers because they can both sense and transmit. These devices work on a principle similar to that of transducers used in radar and sonar systems, which evaluate attributes of a target by interpreting the echoes from radio or sound waves, respectively. Active ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions, convert it to an electrical signal, and report it to a computer.

1.2.8.1 Capabilities and limitations

This technology can be used for measuring wind speed and direction (anemometer), tank or channel level, and speed through air or water. For measuring speed or direction, a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure tank or channel level, the sensor measures the distance to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultrasonography, burglar alarms and non-destructive testing.

Systems typically use a transducer which generates sound waves in the ultrasonic range, above 18 kHz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.

The technology is limited by the shapes of surfaces and the density or consistency of the material. Foam, in particular, can distort surface level readings.

1.2.8.2 Transducers

An ultrasonic transducer is a device that converts energy into ultrasound, or sound waves above the normal range of human hearing. While technically a dog whistle is an ultrasonic transducer that converts mechanical energy in the form of air pressure into ultrasonic sound waves, the term is more apt to be used to refer to piezoelectric transducers or capacitive transducers that convert electrical energy into sound. Piezoelectric crystals have the property of changing size when a voltage is applied; applying an alternating current (AC) across them causes them to oscillate at very high frequencies, thus producing very high frequency sound waves.

The location at which a transducer focuses the sound can be determined by the active transducer area and shape, the ultrasound frequency, and the sound velocity of the propagation medium. The diagrams show the sound fields of an unfocused and a focusing ultrasonic transducer in water.

Since piezoelectric crystals generate a voltage when force is applied to them, the same crystal can be used as an ultrasonic detector. Some systems use separate transmitter and receiver components while others combine both in a single piezoelectric transceiver.

Non-piezoelectric principles are also used in construction of ultrasound transmitters. Magnetostrictive materials slightly change size when exposed to a magnetic field; such materials can be used to make transducers. A capacitor microphone uses a thin plate which moves in response to ultrasound waves; changes in the electric field around the plate convert sound signals to electric currents, which can be amplified.

1.2.8.3 Use in medicine

Medical ultrasonic transducers (probes) come in a variety of different shapes and sizes for use in making pictures of different parts of the body. The transducer may be passed over the surface of the body or inserted into a body opening such as the rectum or vagina. Clinicians who perform ultrasound-guided procedures often use a probe positioning system to hold the ultrasonic transducer.

Air detection sensors are used in various roles. Non-invasive air detection capabilities in the most critical applications where the safety of a patient is mandatory. Many of the variables, which can affect performance of amplitude or continuous wave based sensing systems, are eliminated or greatly reduced, thus yielding accurate and repeatable detection. The principle behind the technology is that the transmit signal consists of short bursts of ultrasonic energy. After each burst, the electronics looks for a return signal within a small window of time corresponding to the time it takes for the energy to pass through the vessel. Only signals received during this period will qualify for additional signal processing.

1.2.8.4 Use in industry

Ultrasonic sensors are used to detect movement of targets and to measure the distance to targets in many automated factories and process plants. Sensors with an on or off digital output are available for detecting the movement of objects, and sensors with an analog output which varies proportionally to the sensor to target separation distance are commercially available. They can be used to sense the edge of material as part of a web guiding system.

Ultrasonic sensors are widely used in automotive applications for parking assist technology. Ultrasonic sensors are being tested in a number of uses including ultrasonic people detection and assisting in autonomous UAV navigation.

Because ultrasonic sensors use sound rather than light for detection, they work in applications where photoelectric sensors may not. Ultrasonics are a great solution for clear object detection, clear label detection and for liquid level measurement, applications that photoelectrics struggle with because of target translucence. Target color and/or reflectivity do not affect ultrasonic sensors which can operate reliably in high-glare environments.

Passive ultrasonic sensors may be used to detect high-pressure gas or liquid leaks, or other hazardous conditions that generate ultrasonic sound.

High-power ultrasonic emitters are used in commercially available ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent (frequently water or isopropanol), and a square wave is applied to it, imparting vibrational energy in the liquid

1.2.9 Wire

A wire is a single, usually cylindrical, flexible strand or rod of metal. Wires are used to bear mechanical loads or electricity and telecommunications signals. Wire is commonly formed by drawing the metal through a hole in a die or draw plate. Wire gauges come in various standard sizes, as expressed in terms of a gauge number. The term *wire* is also used more loosely to refer to a bundle of such strands, as in 'multistranded wire', which is more correctly termed a wire rope in mechanics, or a cable in electricity.

Wire comes in solid core, stranded, or braided forms. Although usually circular in cross-section, wire can be made in square, hexagonal, flattened rectangular, or other cross-sections, either for decorative purposes, or for technical purposes such as high-efficiency voice coils in loudspeakers. Edge-wound coil springs, such as the Slinky toy, are made of special flattened wire.

1.2.9.1 History

In antiquity, jewelry often contains, in the form of chains and applied decoration, large amounts of wire that is accurately made and which must have been produced by some

efficient, if not technically advanced, means. In some cases, strips cut from metal sheet were made into wire by pulling them through perforations in stone beads. This causes the strips to fold round on themselves to form thin tubes. This strip drawing technique was in use in Egypt by the 2nd Dynasty. From the middle of the 2nd millennium BC most of the gold wires in jewellery are characterised by seam lines that follow a spiral path along the wire. Such twisted strips can be converted into solid round wires by rolling them between flat surfaces or the strip wire drawing method. The strip twist wire manufacturing method was superseded by drawing in the ancient Old World sometime between about the 8th and 10th centuries AD. There is some evidence for the use of drawing further East prior to this period.

Square and hexagonal wires were possibly made using a swaging technique. In this method a metal rod was struck between grooved metal blocks, or between a grooved punch and a grooved metal anvil. Swaging is of great antiquity, possibly dating to the beginning of the 2nd millennium BC in Egypt and in the Bronze and Iron Ages in Europe for torcs and fibulae.

Twisted square section wires are a very common filigree decoration in early Etruscan jewellery.

In about the middle of the 2nd millennium BC a new category of decorative tube was introduced which imitated a line of granules. True beaded wire, produced by mechanically distorting a round-section wire, appeared in the Eastern Mediterranean and Italy in the seventh century BC, perhaps disseminated by the Phoenicians. Beaded wire continued to be used in jewellery into modern times, although it largely fell out of favour in about the tenth century AD when two drawn round wires, twisted together to form what are termed 'ropes', provided a simpler-to-make alternative. A forerunner to beaded wire may be the notched strips and wires which first occur from around 2000 BC in Anatolia.

Wire was drawn in England from the medieval period. The wire was used to make wool cards and pins, manufactured goods whose import was prohibited by Edward IV in 1463. The first wire mill in Great Britain was established at Tintern in about 1568 by the founders of the Company of Mineral and Battery Works, who had a monopoly on this. Apart from their second wire mill at nearby Whitebrook, there were no other wire mills

before the second half of the 17th century. Despite the existence of mills, the drawing of wire down to fine sizes continued to be done manually.

Wire is usually drawn of cylindrical form; but it may be made of any desired section by varying the outline of the holes in the draw-plate through which it is passed in the process of manufacture. The draw-plate or die is a piece of hard cast-iron or hard steel, or for fine work it may be a diamond or a ruby. The object of utilising precious stones is to enable the dies to be used for a considerable period without losing their size, and so producing wire of incorrect diameter. Diamond dies must be rebored when they have lost their original diameter of hole, but metal dies are brought down to size again by hammering up the hole and then drifting it out to correct diameter with a punch.

1.2.9.2 Uses

Wire has many uses. It forms the raw material of many important manufacturers, such as the wire netting industry, engineered springs, wire-cloth making and wire rope spinning, in which it occupies a place analogous to a textile fiber. Wire-cloth of all degrees of strength and fineness of mesh is used for sifting and screening machinery, for draining paper pulp, for window screens, and for many other purposes. Vast quantities of aluminium, copper, nickel and steel wire are employed for telephone and data cables, and as conductors in electric power transmission, and heating. It is in no less demand for fencing, and much is consumed in the construction of suspension bridges, and cages, etc. In the manufacture of stringed musical instruments and scientific instruments wire is again largely used. Carbon and stainless spring steel wire have significant applications for engineered springs for critical automotive or industrial manufactured parts/components. Among its other sources of consumption it is sufficient to mention pin and hairpin making, the needle and fish-hook industries, nail, peg and rivet making, and carding machinery; indeed there are few industries into which it does not enter.

Not all metals and metallic alloys possess the physical properties necessary to make useful wire. The metals must in the first place be ductile and strong in tension, the quality on which the utility of wire principally depends. The metals suitable for wire, possessing almost equal ductility, are platinum, silver, iron, copper, aluminium and gold; and it is only from these and certain of their alloys with other metals, principally

brass and bronze, that wire is prepared (*For a detailed discussion on copper wire, see main article: Copper wire and cable.*).

By careful treatment extremely thin wire can be produced. Special purpose wire is however made from other metals (e.g. tungsten wire for light bulb and vacuum tube filaments, because of its high melting temperature). Copper wires are also plated with other metals, such as tin, nickel, and silver to handle different temperatures, provide lubrication, provide easier stripping of rubber from copper.

1.2.9.3 Production

Wire is often reduced to the desired diameter and properties by repeated drawing through progressively smaller dies, or traditionally holes in draw plates. After a number of passes the wire may be annealed to facilitate more drawing or, if it is a finished product, to maximise ductility and conductivity.

1.2.9.4 Finishing, jacketing, and insulating

Electrical wires are usually covered with insulating materials, such as plastic, rubber-like polymers, or varnish. Insulating and jacketing of wires and cables is nowadays done by passing them through an extruder. Formerly, materials used for insulation included treated cloth or paper and various oil-based products. Since the mid-1960s, plastic and polymers exhibiting properties similar to rubber have predominated.

Two or more wires may be wrapped concentrically, separated by insulation, to form coaxial cable. The wire or cable may be further protected with substances like paraffin, some kind of preservative compound, bitumen, lead, aluminum sheathing, or steel taping. Stranding or covering machines wind material onto wire which passes through quickly. Some of the smallest machines for cotton covering have a large drum, which grips the wire and moves it through toothed gears; the wire passes through the centre of disks mounted above a long bed, and the disks carry each a number of bobbins varying from six to twelve or more in different machines. A supply of covering material is wound on each bobbin, and the end is led on to the wire, which occupies a central position relatively to the bobbins; the latter being revolved at a suitable speed bodily with their disks, the cotton is consequently served on to the wire, winding in spiral fashion so as to

overlap. If a large number of strands are required the disks are duplicated, so that as many as sixty spools may be carried, the second set of strands being laid over the first.

For heavier cables that are used for electric light and power as well as submarine cables, the machines are somewhat different in construction. The wire is still carried through a hollow shaft, but the bobbins or spools of covering material are set with their spindles at right angles to the axis of the wire, and they lie in a circular cage which rotates on rollers below. The various strands coming from the spools at various parts of the circumference of the cage all lead to a disk at the end of the hollow shaft. This disk has perforations through which each of the strands pass, thence being immediately wrapped on the cable, which slides through a bearing at this point. Toothed gears having certain definite ratios are used to cause the winding drum for the cable and the cage for the spools to rotate at suitable relative speeds which do not vary. The cages are multiplied for stranding with a large number of tapes or strands, so that a machine may have six bobbins on one cage and twelve on the other.

1.2.9.5 Forms of wire

1.2.9.5.1 Solid wire

Solid wire, also called solid-core or single-strand wire, consists of one piece of metal wire. Solid wire is useful for wiring breadboards. Solid wire is cheaper to manufacture than stranded wire and is used where there is little need for flexibility in the wire. Solid wire also provides mechanical ruggedness; and, because it has relatively less surface area which is exposed to attack by corrosives, protection against the environment.

1.2.9.5.2 Stranded wire

Stranded wire is composed of a number of small gauge wire bundled or wrapped together to form a larger conductor. Stranded wire is more flexible than solid wire of the same total cross-sectional area. Stranded wire tends to be a better conductor than solid wire because the individual wires collectively comprise a greater surface area. Stranded wire is used when higher resistance to metal fatigue is required. Such situations include connections between circuit boards in multi-printed-circuit-board devices, where the rigidity of solid wire would produce too much stress as a result of movement during assembly or servicing; A.C. line cords for appliances; musical instrument cables;

computer mouse cables; welding electrode cables; control cables connecting moving machine parts; mining machine cables; trailing machine cables; and numerous others.

At high frequencies, current travels near the surface of the wire because of the *skin effect*, resulting in increased power loss in the wire. Stranded wire might seem to reduce this effect, since the total surface area of the strands is greater than the surface area of the equivalent solid wire, but ordinary stranded wire does not reduce the skin effect because all the strands are short-circuited together and behave as a single conductor. A stranded wire will have higher resistance than a solid wire of the same diameter because the cross-section of the stranded wire is not all copper; there are unavoidable gaps between the strands (this is the circle packing problem for circles within a circle). A stranded wire with the same cross-section of conductor as a solid wire is said to have the same equivalent gauge and is always a larger diameter.

However, for many high-frequency applications, *proximity effect* is more severe than skin effect, and in some limited cases, simple stranded wire can reduce proximity effect. For better performance at high frequencies, litz wire, which has the individual strands insulated and twisted in special patterns, may be used.

1.2.9.5.3 Braided wire

A braided wire is composed of a number of small strands of wire braided together. Similar to stranded wires, braided wires are better conductors than solid wires. Braided wires do not break easily when flexed. Braided wires are often suitable as an electromagnetic shield in noise-reduction cables.

1.2.9.5.4 Number of strands

The more individual wire strands in a wire bundle, the more flexible, kink-resistant, break-resistant, and stronger the wire is. But more strands increase cost.

The lowest number of strands usually seen is 7: one in the middle, 6 surrounding it. The next level up is 19, which is another layer of 12 strands on top of the 7. After that the number varies, but 37 and 49 are common, then in the 70 to 100 range (the number is no longer exact). Even larger numbers than that are typically found only in very large cables.

For application where the wire moves, 19 is the lowest that should be used (7 should only be used in applications where the wire is placed and then does not move), and 49 is much better. For applications with constant repeated movement, such as assembly robots and headphone wires, 70 to 100 is mandatory.

For applications that need even more flexibility (welding is the usual example, but also any need to move wire in tight areas), even more strands are used. One example is a 2/0 wire made from 5,292 strands of #36 gauge wire. The strands are organized by first creating a bundle of 7 strands. Then 7 of these bundles are put together into super bundles. Finally 108 super bundles are used to make the final cable. Each group of wires is wound in a helix so that when the wire is flexed, the part of a bundle that is stretched moves around the helix to a part that is compressed to allow the wire to have less stress.

1.2.9.6 Varieties

- ***Hook-up wire* is small-to-medium gauge, solid or stranded, insulated wire, used for making internal connections inside electrical or electronic devices. It is often tin-plated to facilitate soldering.**
- ***Wire bonding* is the application of microscopic wires for making electrical connections inside semiconductor components and integrated circuits.**
- ***Magnet wire* is solid wire, usually copper, which, to allow closer winding when making electromagnetic coils, is insulated only with varnish, rather than the thicker plastic or other insulation commonly used on electrical wire. It is used for the winding of motors, transformers, inductors, generators, speaker coils, *etc.* (*For further information about copper magnet wire, see: Copper wire and cable#Magnet wire (Winding wire)*.).**
- ***Coaxial cable* is a cable consisting of an inner conductor, surrounded by a tubular insulating layer typically made from a flexible material with a high dielectric constant, all of which is then surrounded by another conductive layer (typically of fine woven wire for flexibility, or of a thin metallic foil), and then finally covered again with a thin insulating layer on the outside. The term coaxial comes from the inner conductor and the outer shield sharing the same geometric axis. Coaxial cables are often used as a transmission line for radio frequency signals. In a hypothetical ideal coaxial cable the electromagnetic field carrying the signal exists only in**

the space between the inner and outer conductors. Practical cables achieve this objective to a high degree. A Coaxial Cable provides protection of signals from external electromagnetic interference, and effectively guides signals with low emission along the length of the cable.

- ***Speaker wire* is used to make the electrical connection between loudspeakers and audio amplifiers. Modern speaker wire consists of electrical conductors individually insulated by plastic.**
- ***Resistance wire* is wire with higher than normal resistivity, often used for heating elements or for making wire-wound resistors. Nichrome wire is the most common type.**

1.2.10 Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

1.2.10.1 Mechanical

A joy buzzer is an example of a purely mechanical buzzer. They require a driver.

1.2.10.2 Electromechanical

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

1.2.10.3 Piezoelectric

1.2.10.4 A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

1.2.10.5 Uses

- **Annunciator panels**
- **Electronic metronomes**
- **Game show lock-out device**
- **Microwave ovens and other household appliances**
- **Sporting events such as basketball games**
- **Electrical alarms**

1.2.11 Electrical resistance and conductance

The electrical resistance of an electrical conductor is the opposition to the passage of an electric current through that conductor. The inverse quantity is electrical conductance, the ease with which an electric current passes. Electrical resistance shares some conceptual parallels with the notion of mechanical friction. The SI unit of electrical resistance is the ohm (Ω), while electrical conductance is measured in siemens (S).

An object of uniform cross section has a resistance proportional to its resistivity and length and inversely proportional to its cross-sectional area. All materials show some resistance, except for superconductors, which have a resistance of zero.

The resistance (R) of an object is defined as the ratio of voltage across it (*V*) to current through it (*I*), while the conductance (G) is the inverse:

$$R = \frac{V}{I}, \qquad G = \frac{I}{V} = \frac{1}{R}$$

For a wide variety of materials and conditions, *V* and *I* are directly proportional to each other, and therefore *R* and *G* are constant (although they can depend on other factors like temperature or strain). This proportionality is called Ohm's law, and materials that satisfy it are called "Ohmic" materials.

In other cases, such as a diode or battery, *V* and *I* are *not* directly proportional, or in other words the *I*–*V* curve is not a straight line through the origin, and Ohm's law does not hold. In this case, resistance and conductance are less useful concepts, and more difficult to define. The ratio V/I is sometimes still useful, and is referred to as a "chordal resistance" or "static resistance", as it corresponds to the inverse slope of a chord between the origin and an *I*–*V* curve. In other situations, the derivative $\frac{dV}{dI}$ may be most useful; this is called the "differential resistance".

1.2.11.1 Introduction

In the hydraulic analogy, current flowing through a wire (or resistor) is like water flowing through a pipe, and the voltage drop across the wire is like the pressure drop that pushes water through the pipe. Conductance is proportional to how much flow occurs for a given pressure, and resistance is proportional to how much pressure is required to achieve a given flow. (Conductance and resistance are reciprocals.)

The voltage *drop* (i.e., difference between voltages on one side of the resistor and the other), not the voltage itself, provides the driving force pushing current through a resistor. In hydraulics, it is similar: The pressure *difference* between two sides of a pipe, not the pressure itself, determines the flow through it. For example, there may be a large water pressure above the pipe, which tries to push water down through the pipe. But there may be an equally large water pressure below the pipe, which tries to push water back up through the pipe. If these pressures are equal, no water flows. (In the image at right, the water pressure below the pipe is zero.)

The resistance and conductance of a wire, resistor, or other element is mostly determined by two properties:

- **geometry (shape), and**
- **material**

Geometry is important because it is more difficult to push water through a long, narrow pipe than a wide, short pipe. In the same way, a long, thin copper wire has higher resistance (lower conductance) than a short, thick copper wire.

Materials are important as well. A pipe filled with hair restricts the flow of water more than a clean pipe of the same shape and size. Similarly, electrons can flow freely and easily through a copper wire, but cannot flow as easily through a steel wire of the same shape and size, and they essentially cannot flow at all through an insulator like rubber, regardless of its shape. The difference between, copper, steel, and rubber is related to their microscopic structure and electron configuration, and is quantified by a property called resistivity.

In addition to geometry and material, there are various other factors that influence resistance and conductance, such as temperature; see below.

1.2.11.2 Conductors and resistors

Substances in which electricity can flow are called conductors. A piece of conducting material of a particular resistance meant for use in a circuit is called a resistor. Conductors are made of high-conductivity materials such as metals, in particular copper and aluminium. Resistors, on the other hand, are made of a wide variety of materials depending on factors such as the desired resistance, amount of energy that it needs to dissipate, precision, and costs.

1.2.11.3 Ohm's law

Ohm's law is an empirical law relating the voltage *V* across an element to the current *I* through it:

$$V \propto I$$

(*V* is directly proportional to *I*). This law is not always true: For example, it is false for diodes, batteries, etc. However, it is true to a very good approximation for wires and resistors (assuming that other conditions, including temperature, are held constant). Materials or objects where Ohm's law is true are called *ohmic*, whereas objects that do not obey Ohm's law are *non-ohmic*.

1.2.11.4 Relation to resistivity and conductivity

The resistance of a given object depends primarily on two factors: What material it is made of, and its shape. For a given material, the resistance is inversely proportional to the cross-sectional area; for example, a thick copper wire has lower resistance than an otherwise-identical thin copper wire. Also, for a given material, the resistance is proportional to the length; for example, a long copper wire has higher resistance than an otherwise-identical short copper wire. The resistance R and conductance G of a conductor of uniform cross section, therefore, can be computed as

$$R = \rho \frac{\ell}{A},$$

$$G = \sigma \frac{A}{\ell}.$$

where ℓ is the length of the conductor, measured in metres [m], ***A*** **is the cross-sectional area of the conductor measured in square metres [m²], σ (sigma) is the electrical conductivity measured in siemens per meter ($S \cdot m^{-1}$), and ρ (rho) is the electrical resistivity (also called** ***specific electrical resistance*****) of the material, measured in ohm-metres (Ω·m). The resistivity and conductivity are proportionality constants, and therefore depend only on the material the wire is made of, not the geometry of the wire. Resistivity and conductivity are reciprocals: $\rho = 1/\sigma$. Resistivity is a measure of the material's ability to oppose electric current.**

This formula is not exact, as it assumes the current density is totally uniform in the conductor, which is not always true in practical situations. However, this formula still provides a good approximation for long thin conductors such as wires.

Another situation for which this formula is not exact is with alternating current (AC), because the skin effect inhibits current flow near the center of the conductor. For this reason, the ***geometrical*** **cross-section is different from the** ***effective*** **cross-section in which current actually flows, so resistance is higher than expected. Similarly, if two conductors near each other carry AC current, their resistances increase due to the proximity effect. At commercial power frequency, these effects are significant for large conductors**

carrying large currents, such as busbars in an electrical substation, or large power cables carrying more than a few hundred amperes.

1.2.11.4.1 What determines resistivity?

Main article: Electrical resistivity and conductivity

The resistivity of different materials varies by an enormous amount: For example, the conductivity of teflon is about 10^{30} times lower than the conductivity of copper. Why is there such a difference? Loosely speaking, a metal has large numbers of "delocalized" electrons that are not stuck in any one place, but free to move across large distances, whereas in an insulator (like teflon), each electron is tightly bound to a single molecule, and a great force is required to pull it away. Semiconductors lie between these two extremes. More details can be found in the article: Electrical resistivity and conductivity. For the case of electrolyte solutions, see the article: Conductivity (electrolytic).

Resistivity varies with temperature. In semiconductors, resistivity also changes when exposed to light. See below.

1.2.11.5 Measuring resistance

Main article: ohmmeter

An instrument for measuring resistance is called an ohmmeter. Simple ohmmeters cannot measure low resistances accurately because the resistance of their measuring leads causes a voltage drop that interferes with the measurement, so more accurate devices use four-terminal sensing.

1.2.11.6 Typical resistances

See also: Electrical resistivities of the elements (data page) and Electrical resistivity and conductivity

1.2.11.7 Static and differential resistance

Many electrical elements, such as diodes and batteries do *not* satisfy Ohm's law. These are called *non-ohmic* or *nonlinear*, and are characterized by an *I–V* curve, which is *not* a straight line through the origin.

Resistance and conductance can still be defined for non-ohmic elements. However, unlike ohmic resistance, nonlinear resistance is not constant but varies with the voltage or current through the device, i.e. its operating point. There are two types of resistance:

- **Static resistance (also called *chordal* or *DC resistance*) - This corresponds to the usual definition of resistance; the voltage divided by the current**

$$R_{\text{static}} = \frac{V}{I}.$$

It is the slope of the line (chord) from the origin through the point on the curve. Static resistance determines the power dissipation in an electrical component. Points on the *IV* curve located in the 2nd or 4th quadrants, for which the slope of the chordal line is negative, have *negative static resistance*. Passive devices, which have no source of energy, cannot have negative static resistance. However active devices such as transistors or op-amps can synthesize negative static resistance with feedback, and it is used in some circuits such as gyrators.

- **Differential resistance (also called *dynamic*, *incremental* or *small signal resistance*) - Differential resistance is the derivative of the voltage with respect to the current; the slope of the *IV* curve at a point**

$$R_{\text{diff}} = \frac{dV}{dI}.$$

If the *IV* curve is nonmonotonic (with peaks and troughs), the curve has a negative slope in some regions—so in these regions the device has *negative differential resistance*. Devices with negative differential resistance can amplify a signal applied to them, and are used to make amplifiers and oscillators. These include tunnel diodes, Gunn diodes, IMPATT diodes, magnetron tubes, and unijunction transistors.

1.2.11.8 AC circuits

1.2.11.8.1 Impedance and admittance

When an alternating current flows through a circuit, the relation between current and voltage across a circuit element is characterized not only by the ratio of their magnitudes, but also the difference in their phases. For example, in an ideal resistor, the moment when the voltage reaches its maximum, the current also reaches its maximum (current and voltage are oscillating in phase). But for a capacitor or inductor, the

maximum current flow occurs as the voltage passes through zero and vice versa (current and voltage are oscillating 90° out of phase, see image at right). Complex numbers are used to keep track of both the phase and magnitude of current and voltage:

$$V(t) = \mathrm{Re}(V_0 e^{j\omega t}), \quad I(t) = \mathrm{Re}(I_0 e^{j\omega t}), \quad Z = \frac{V_0}{I_0}, \quad Y = \frac{I_0}{V_0}$$

where:

- ***t* is time,**
- ***V*(*t*) and *I*(*t*) are, respectively, voltage and current as a function of time,**
- **V_0, I_0, *Z*, and *Y* are complex numbers,**
- ***Z* is called impedance,**
- ***Y* is called admittance,**
- **Re indicates real part,**
- **ω is the angular frequency of the AC current,**
- **$j = \sqrt{-1}$ is the imaginary unit.**

The impedance and admittance may be expressed as complex numbers that can be broken into real and imaginary parts:

$$Z = R + jX, \quad Y = G + jB$$

where *R* and *G* are resistance and conductance respectively, *X* is reactance, and *B* is susceptance. For ideal resistors, *Z* and *Y* reduce to *R* and *G* respectively, but for AC networks containing capacitors and inductors, *X* and *B* are nonzero.

$Z = 1/Y$ for AC circuits, just as $R = 1/G$ for DC circuits.

1.2.11.8.2 Frequency dependence of resistance

Another complication of AC circuits is that the resistance and conductance can be frequency-dependent. One reason, mentioned above is the skin effect (and the related proximity effect). Another reason is that the resistivity itself may depend on frequency (see Drude model, deep-level traps, resonant frequency, Kramers–Kronig relations, etc.)

1.2.11.9 Energy dissipation and Joule heating

Resistors (and other elements with resistance) oppose the flow of electric current; therefore, electrical energy is required to push current through the resistance. This electrical energy is dissipated, heating the resistor in the process. This is called *Joule heating* (after James Prescott Joule), also called *ohmic heating* or *resistive heating*.

The dissipation of electrical energy is often undesired, particularly in the case of transmission losses in power lines. High voltage transmission helps reduce the losses by reducing the current for a given power.

On the other hand, Joule heating is sometimes useful, for example in electric stoves and other electric heaters (also called *resistive heaters*). As another example, incandescent lamps rely on Joule heating: the filament is heated to such a high temperature that it glows "white hot" with thermal radiation (also called incandescence).

The formula for Joule heating is:

$$P = I^2 R$$

where P is the power (energy per unit time) converted from electrical energy to thermal energy, R is the resistance, and I is the current through the resistor.

1.2.11.10 Dependence of resistance on other conditions

1.2.11.10.1 Temperature dependence

Main article: Electrical resistivity and conductivity § Temperature dependence

Near room temperature, the resistivity of metals typically increases as temperature is increased, while the resistivity of semiconductors typically decreases as temperature is increased. The resistivity of insulators and electrolytes may increase or decrease depending on the system. For the detailed behavior and explanation, see Electrical resistivity and conductivity.

As a consequence, the resistance of wires, resistors, and other components often change with temperature. This effect may be undesired, causing an electronic circuit to malfunction at extreme temperatures. In some cases, however, the effect is put to good

use. When temperature-dependent resistance of a component is used purposefully, the component is called a resistance thermometer or thermistor. (A resistance thermometer is made of metal, usually platinum, while a thermistor is made of ceramic or polymer.)

Resistance thermometers and thermistors are generally used in two ways. First, they can be used as thermometers: By measuring the resistance, the temperature of the environment can be inferred. Second, they can be used in conjunction with Joule heating (also called self-heating): If a large current is running through the resistor, the resistor's temperature rises and therefore its resistance changes. Therefore, these components can be used in a circuit-protection role similar to fuses, or for feedback in circuits, or for many other purposes. In general, self-heating can turn a resistor into a nonlinear and hysteretic circuit element. For more details see Thermistor#Self-heating effects.

If the temperature *T* does not vary too much, a linear approximation is typically used:

$$R(T) = R_0[1 + \alpha(T - T_0)]$$

where α is called the *temperature coefficient of resistance*, T_0 is a fixed reference temperature (usually room temperature), and R_0 is the resistance at temperature T_0. The parameter α is an empirical parameter fitted from measurement data. Because the linear approximation is only an approximation, α is different for different reference temperatures. For this reason it is usual to specify the temperature that α was measured at with a suffix, such as α_{15}, and the relationship only holds in a range of temperatures around the reference.

The temperature coefficient α is typically $+3\times10^{-3}$ K^{-1} to $+6\times10^{-3}$ K^{-1} for metals near room temperature. It is usually negative for semiconductors and insulators, with highly variable magnitude.

1.2.11.10.2 Strain dependence

Main article: Strain gauge

Just as the resistance of a conductor depends upon temperature, the resistance of a conductor depends upon strain. By placing a conductor under tension (a form of stress that leads to strain in the form of stretching of the conductor), the length of the section of conductor under tension increases and its cross-sectional area decreases. Both these

effects contribute to increasing the resistance of the strained section of conductor. Under compression (strain in the opposite direction), the resistance of the strained section of conductor decreases. See the discussion on strain gauges for details about devices constructed to take advantage of this effect.

1.2.11.10.3 Light illumination dependence

Main articles: Photoresistor and Photoconductivity

Some resistors, particularly those made from semiconductors, exhibit *photoconductivity*, meaning that their resistance changes when light is shining on them. Therefore they are called *photoresistors* (or *light dependent resistors*). These are a common type of light detector.

1.2.11.11 Superconductivity

Main article: Superconductivity

Superconductors are materials that have exactly zero resistance and infinite conductance, because they can have V=0 and I≠0. This also means there is no joule heating, or in other words no dissipation of electrical energy. Therefore, if superconductive wire is made into a closed loop, current flows around the loop forever. Superconductors require cooling to temperatures near 4 K with liquid helium for most metallic superconductors like NbSn alloys, or cooling to temperatures near 77K with liquid nitrogen for the expensive, brittle and delicate ceramic high temperature superconductors. Nevertheless, there are many technological applications of superconductivity, including superconducting magnets.

1.2.12 LED lamp

An LED lamp is a light-emitting diode (LED) product that is assembled into a *lamp* (or *light bulb*) for use in lighting fixtures. LED lamps have a lifespan and electrical efficiency that is several times better than incandescent lamps, and significantly better than most fluorescent lamps, with some chips able to emit more than 100 lumens per watt. The LED lamp market is projected to grow by more than twelve-fold over the next

decade, from $2 billion in the beginning of 2014 to $25 billion in 2023, a compound annual growth rate (CAGR) of 25%.

Like incandescent lamps and unlike most fluorescent lamps (e.g. tubes and compact fluorescent lamps or CFLs), LEDs come to full brightness without need for a warm-up time; the life of fluorescent lighting is also reduced by frequent switching on and off.[*citation needed*] **Initial cost of LED is usually higher. Degradation of LED dye and packaging materials reduces light output to some extent over time.**

Some LED lamps are made to be a directly compatible drop-in replacement for incandescent or fluorescent lamps. An LED lamp packaging may show the lumen output, power consumption in watts, color temperature in kelvins or description (e.g. "warm white"), operating temperature range, and sometimes the equivalent wattage of an incandescent lamp of similar luminous output.

LEDs do not emit light in all directions, and their directional characteristics affect the design of lamps. The light output of single LEDs is less than that of incandescent and compact fluorescent lamps; in most applications multiple LEDs are used to form a lamp, although high-power versions (see below) are becoming available.

LED chips need controlled direct current (DC) electrical power; an appropriate circuit is required to convert alternating current from the supply to the regulated low voltage direct current used by the LEDs. LEDs are adversely affected by high temperature, so LED lamps typically include heat dissipation elements such as heat sinks and cooling fins.

1.2.12.1 Technology overview

General-purpose lighting needs white light. LEDs emit light in a very narrow band of wavelengths, emitting light of a color characteristic of the energy bandgap of the semiconductor material used to make the LED. To emit white light from LEDs requires either mixing light from red, green, and blue LEDs, or using a phosphor to convert some of the light to other colors.

One method (RGB or trichromatic white LEDs) uses multiple LED chips, each emitting a different wavelength, in close proximity to generate white light. This allows the intensity of each LED to be adjusted to change the overall color.

The second method uses LEDs in conjunction with a phosphor. The CRI (color rendering index) value can range from less than 70 to over 90, and color temperatures in the range of 2700 K (matching incandescent lamps) up to 7000 K are available.[*citation needed*]

1.2.12.2 Application

A significant difference from other light sources is that the light is more directional, i.e., emitted as a narrower beam. LED lamps are used for both general and special-purpose lighting. Where colored light is needed, LEDs that inherently emit light of a single color require no energy-absorbing filters.

White-light LED lamps have longer life expectancy and higher efficiency (more light for the same electricity) than most other lighting when used at the proper temperature. LED sources are compact, which gives flexibility in designing lighting fixtures and good control over the distribution of light with small reflectors or lenses. Because of the small size of LEDs, control of the spatial distribution of illumination is extremely flexible, and the light output and spatial distribution of an LED array can be controlled with no efficiency loss.

LEDs using the color-mixing principle can emit a wide range of colors by changing the proportions of light generated in each primary color. This allows full color mixing in lamps with LEDs of different colors. Unlike other lighting technologies, LED emission tends to be directional (or at least lambertian), which can be either advantageous or disadvantageous, depending on requirements. For applications where non-directional light is required, either a diffuser is used, or multiple individual LED emitters are used to emit in different directions.

1.2.12.3 Household LED lamps

1.2.12.3.1 Replacement for existing lighting

1.2.12.3.2 Lamp sizes and bases

LED lamps are made of arrays of SMD modules that replace screw-in incandescent or compact fluorescent light bulbs, mostly replacing incandescent bulbs rated from 5 to 60 watts. Such lamps are made with standard light bulb connections and shapes, such as an Edison screw base, an MR16 shape with a bi-pin base, or a GU5.3 (bi-pin cap) or GU10 (bayonet fitting) and are made compatible with the voltage supplied to the sockets. They include driver circuitry to rectify the AC power and convert the voltage to an appropriate value, usually Switched-mode power supplies.

As of 2010 some LED lamps replaced higher wattage bulbs; for example, one manufacturer claimed a 16-watt LED bulb was as bright as a 150 W halogen lamp. A standard general-purpose incandescent bulb emits light at an efficiency of about 14 to 17 lumens/W depending on its size and voltage. According to the European Union standard, an energy-efficient bulb that claims to be the equivalent of a 60 W tungsten bulb must have a minimum light output of 806 lumens

Some models of LED bulbs are compatible with dimmers as used for incandescent lamps. LED lamps often have directional light characteristics. The lamps have declined in cost to between US $10 to $50 each as of 2012. These bulbs are more power-efficient than compact fluorescent bulbs and offer lifespans of 30,000 or more hours, reduced if operated at a higher temperature than specified. Incandescent bulbs have a typical life of 1,000 hours, and compact fluorescents about 8,000 hours. The bulbs maintain output light intensity well over their lifetimes. Energy Star specifications require the bulbs to typically drop less than 10% after 6,000 or more hours of operation, and in the worst case not more than 15%. LED lamps are available with a variety of color properties. The purchase price is higher than most other, but the higher efficiency may make total cost of ownership (purchase price plus cost of electricity and changing bulbs) lower

Several companies offer LED lamps for general lighting purposes. The technology is improving rapidly and new energy-efficient consumer LED lamps are available.

LED lamps are close to being adopted as the mainstream light source because of the falling prices and because 40 and 60 watt incandescent bulbs are being phased out. In

the U.S. the Energy Independence and Security Act of 2007 effectively bans the manufacturing and importing of most current incandescent light bulbs. LED bulbs have decreased substantially in pricing and many varieties are sold with subsidized prices from local utilities. At some Wal-Mart locations prices for LED bulbs have been under $1

1.2.12.3.3 LED tube lamps

LED tube lights are designed to physically fit in fixtures intended for fluorescent tubes. Some LED tube lamps are intended to be a drop-in replacement into existing fixtures. Others require rewiring of the fixtures to remove the ballast. An LED tube lamp generally uses many individual LEDs which are directional. Fluorescent lamps emit light all the way around the lamp. Most LED tube lights available can be used in place of T8, T10, or T12 tube designations, in lengths of 2, 4, 6, and 8 feet.

1.2.12.3.4 Lighting designed for LEDs

Newer light fittings designed for LED lamps, or indeed with long-lived LEDs built-in, have been coming into use as the need for compatibility with existing fittings diminishes. Such lighting does not require each bulb to contain circuitry to operate from mains voltage.

1.2.12.4 Specialty uses

White LED lamps have achieved market dominance in applications where high efficiency is important at low power levels. Some of these applications include flashlights, solar-powered garden or walkway lights, and bicycle lights. Monochromatic (colored) LED lamps are now commercially used for traffic signal lamps, where the ability to emit bright monochromatic light is a desired feature, and in strings of holiday lights.

1.2.12.5 Comparison to other lighting technologies

See luminous efficacy for an efficiency chart comparing various technologies.

- **Incandescent lamps (light bulbs) generate light by passing electric current through a resistive filament, thereby heating the filament to a very high** temperature so that it glows and emits visible light over a broad range of wavelengths. Incandescent sources yield a "warm" yellow or white color quality depending on the filament operating temperature. Incandescent lamps emit 98% of the energy input as heat. A 100 W light bulb for 120 V operation emits about 1,700 lumens, about 17 lumens/W;for 230 V bulbs the figures are 1340 lm and 13.4 lm/W. Incandescent lamps are relatively inexpensive to make. The typical lifespan of an AC incandescent lamp is 750 to 1,000 hours. They work well with dimmers. Most older light fixtures are designed for the size and shape of these traditional bulbs. In the U.S. the regular sockets are E26 and E11, and E27 and E14 in some European countries.

- Fluorescent lamps work by passing electricity through mercury vapor, which in turn emits ultraviolet light. The ultraviolet light is then absorbed by a phosphor coating inside the lamp, causing it to glow, or fluoresce. Conventional linear fluorescent lamps have life spans around 20,000 and 30,000 hours based on 3 hours per cycle according to lamps NLPIP reviewed in 2006. Induction fluorescent relies on electromagnetism rather than the cathodes used to start conventional linear fluorescent. The newer rare earth triphosphor blend linear fluorescent lamps made by Osram, Philips, Crompton and others have a life expectancy greater than 40,000 hours, if coupled with a warm-start electronic ballast. The life expectancy depends on the number of on/off cycles, and is lower if the light is cycled often. The ballast-lamp combined system efficacy for then current linear fluorescent systems in 1998 as tested by NLPIP ranged from 80 to 90 lm/W. For comparison, general household LED bulbs available in 2011 emit 64 lumens/W.

- Compact fluorescent lamps' specified lifespan typically ranges from 6,000 hours to 15,000 hours.

- Electricity prices vary state to state and are customer dependent. Generally commercial (10.3 cent/kWh) and industrial (6.8 cent/kWh) electricity prices are lower than residential (12.3 cent/kWh) due to fewer transmission losses

In keeping with the long life claimed for LED lamps, long warranties are offered. One manufacturer warrants lamps for professional use, depending upon type, for periods of (defined) "normal use" ranging from 1 year or 2,000 hours (whichever comes first) to 5 years or 20,000 hours. A typical domestic LED lamp is stated to have an "average life" of 15,000 hours (15 years at 3 hours/day), and to support 50,000 switch cycles.

1.2.12.5.1 Energy Star qualification

Energy Star is an international standard for energy efficient consumer products. Devices carrying the Energy Star service mark generally use 20–30% less energy than required by US standards.

Energy Star LED qualifications:

- **Reduces energy costs — uses at least 75% less energy than incandescent lighting, saving on operating expenses.**
- **Reduces maintenance costs — lasts 35 to 50 times longer than incandescent lighting and about 2 to 5 times longer than fluorescent lighting. No bulb-replacements, no ladders, no ongoing disposal program.**
- **Reduces cooling costs — LEDs produce very little heat.**
- **Is guaranteed — comes with a minimum three-year warranty — far beyond the industry standard.**
- **Offers convenient features — available with dimming on some indoor models and automatic daylight shut-off and motion sensors on some outdoor models.**
- **Is durable — won't break like a bulb.**

To qualify for Energy Star certification, LED lighting products must pass a variety of tests to prove that the products will display the following characteristics:

- **Brightness is equal to or greater than existing lighting technologies (incandescent or fluorescent) and light is well distributed over the area lighted by the fixture.**
- **Light output remains constant over time, only decreasing towards the end of the rated lifetime (at least 35,000 hours or 12 years based on use of 8 hours per day).**
- **Excellent color quality. The shade of white light appears clear and consistent over time.**
- **Efficiency is as good as or better than fluorescent lighting.**
- **Light comes on instantly when turned on.**

- **No flicker when dimmed.**
- **No off-state power draw. The fixture does not use power when it is turned off, with the exception of external controls, whose power should not exceed 0.5 watts in the off state.**

1.2.12.6 Limitations

Color rendition is not identical to incandescent lamps. A measurement unit called CRI is used to express how the light source's ability to render the eight color sample chips compare to a reference on a scale from 0 to 100. LEDs with CRI below 75 are not recommended for use in indoor lighting.

LED efficiency and life span drop at higher temperatures, which limits the power that can be used in lamps that physically replace existing filament and compact fluorescent types. Thermal management of high-power LEDs is a significant factor in design of solid state lighting equipment.

LED lamps are sensitive to excessive heat, like most solid state electronic components. LED lamps should be checked for compatibility for use in totally or partially enclosed fixtures before installation since heat build-up could cause lamp failure and/or fire.

LED lamps may flicker. The extent of flicker is based on the quality of the DC power supply built into the lamp structure, usually located in the lamp base.

Depending on the design of the lamp, the LED lamp may be sensitive to electrical surges.This is generally not an issue with incandescents, but can be an issue with LED and compact fluorescent bulbs. Power circuits that supply LED lamps can be protected from electrical surges through the use of surge protection devices.

The long life of LEDs, expected to be about 50 times that of the most common incandescent bulbs and significantly longer than fluorescent types, is advantageous for users but will affect manufacturers as it reduces the market for replacements in the distant future.

1.2.12.6.1 Efficiency droop

The term "efficiency droop" refers to the decrease in luminous efficacy of LEDs as the electrical current increases above tens of milliamps (mA). Instead of increasing current levels, luminance is usually increased by combining multiple LEDs in one bulb. Solving the problem of efficiency droop would mean that household LED light bulbs would need fewer LEDs, which would significantly reduce costs.

In addition to being less efficient, operating LEDs at higher electrical currents creates higher heat levels which compromise the lifetime of the LED. Because of this increased heating at higher currents, high-brightness LEDs have an industry standard of operating at only 350 mA. 350 mA is a good compromise between light output, efficiency, and longevity.

Early suspicions were that the LED droop was caused by elevated temperatures. Scientists proved the opposite to be true that, although the life of the LED would be shortened, elevated temperatures actually improved the efficiency of the LED. The mechanism causing efficiency droop was identified in 2007 as Auger recombination, which was taken with mixed reaction. In 2013, a study conclusively identified Auger recombination as the cause of efficiency droop.

1.2.12.7 Development and adoption history

The first LEDs were developed in the early 1960s, however, they were low-powered and only produced light in the low, red frequencies of the spectrum. The first high-brightness blue LED was demonstrated by Shuji Nakamura of Nichia Corporation in 1994. The existence of blue LEDs and high-efficiency LEDs quickly led to the development of the first white LED, which employed a phosphor coating to mix down-converted yellow light with blue to produce light that appears white. Isamu Akasaki, Hiroshi Amano and Nakamura were later awarded the 2014 Nobel prize in physics for the invention of the blue LED.

The Energy Independence and Security Act (EISA) of 2007 authorized the Department of Energy (DOE) to establish the Bright Tomorrow Lighting Prize competition, known as the "L Prize", the first government-sponsored technology competition designed to challenge industry to develop replacements for 60 W incandescent lamps and PAR 38

halogen lamps. The EISA legislation established basic requirements and prize amounts for each of the two competition categories, and authorized up to $20 million in cash prizes. The competition also included the possibility for winners to obtain federal purchasing agreements, utility programs, and other incentives. In May 2008, they announced details of the competition and technical requirements for each category. Lighting products meeting the competition requirements could use just 17% of the energy used by most incandescent lamps in use today. That same year the DOE also launched the Energy Star program for solid-state lighting products. The EISA legislation also authorized an additional L Prize program for developing a new "21st Century Lamp".

Philips Lighting ceased research on compact fluorescents in 2008 and began devoting the bulk of its research and development budget to solid-state lighting. On 24 September 2009, Philips Lighting North America became the first to submit lamps in the category to replace the standard 60 W A-19 "Edison screw fixture" light bulb, with a design based on their earlier "AmbientLED" consumer product. On 3 August 2011, DOE awarded the prize in the 60 W replacement category to a Philips' LED lamp after 18 months of extensive testing.

Early LED lamps varied greatly in chromaticity from the incandescent lamps they were replacing. A standard was developed, ANSI C78.377-2008, that specified the recommended color ranges for solid-state lighting products using cool to warm white LEDs with various correlated color temperatures. In June 2008, NIST announced the first two standards for solid-state lighting in the United States. These standards detail performance specifications for LED light sources and prescribe test methods for solid-state lighting products.

Also in 2008 in the United States and Canada, the Energy Star program began to label lamps that meet a set of standards for starting time, life expectancy, color, and consistency of performance. The intent of the program is to reduce consumer concerns due to variable quality of products, by providing transparency and standards for the labeling and usability of products available in the market. Energy Star Light Bulbs for Consumers is a resource for finding and comparing Energy Star qualified lamps. A similar program in the United Kingdom (run by the Energy Saving Trust) was launched to identify lighting products that meet energy conservation and performance guidelines.

The Illuminating Engineering Society of North America (IESNA) published a documentary standard LM-79, which describes the methods for testing solid-state lighting products for their light output (lumens), efficacy (lumens per watt) and chromaticity.

In January 2009, it was reported that researchers at Cambridge University had developed an LED bulb that costs £2 (about $3 U.S.), is 12 times as energy efficient as a tungsten bulb, and lasts for 100,000 hours. Honeywell Electrical Devices and Systems (ED&S) recommend world wide usage of LED lighting as it is energy efficient and can help save the climate.

1.2.12.7.1 Examples of early adoption

In 2008 Sentry Equipment Corporation in Oconomowoc, Wisconsin, USA, was able to light its new factory interior and exterior almost solely with LEDs. Initial cost was three times more than a traditional mix of incandescent and fluorescent lamps, but the extra cost was recovered within two years via electricity savings, and the lamps should not need replacing for 20 years. In 2009 the Manapakkam, Chennai office of the Indian IT company, iGate, spent 3,700,000 (US$80,000) to light 57,000 sq ft (5,300 m^2) of office space with LEDs. The firm expected the new lighting to pay for itself fully within 5 years.

In 2009 the exceptionally large Christmas tree standing in front of the Turku Cathedral in Finland was hung with 710 LED bulbs, each using 2 watts. It has been calculated that these LED lamps paid for themselves in three and a half years, even though the lights run for only 48 days per year.

In 2009 a new highway (A29) was inaugurated in Aveiro, Portugal, it included the first European public LED-based lighting highway.

By 2010 mass installations of LED lighting for commercial and public uses were becoming common. LED lamps were used for a number of demonstration projects for outdoor lighting and LED street lights. The United States Department of Energy made several reports available on the results of many pilot projects for municipal outdoor lighting, and many additional streetlight and municipal outdoor lighting projects soon followed

1.2.13 Breadboard

A breadboard (or protoboard) is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (AKA plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these. "Breadboard" is also a synonym for "prototype".

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. A stripboard (veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units

1.2.13.1 Evolution

In the early days of radio, amateurs nailed bare copper wires or terminal strips to a wooden board (often literally a board to slice bread on) and soldered electronic components to them. Sometimes a paper schematic diagram was first glued to the board as a guide to placing terminals, then components and wires were installed over their symbols on the schematic. Using thumbtacks or small nails as mounting posts was also common.

Breadboards have evolved over time, with the term now being used for all kinds of prototype electronic devices. For example, US Patent 3,145,483, filed in 1961 and granted in 1964, describes a wooden plate breadboard with mounted springs and other facilities. US Patent 3,496,419,filed in 1967 and granted in 1970, refers to a particular printed circuit board layout as a *Printed Circuit Breadboard*. Both examples refer to and describe other types of breadboards as prior art.

The breadboard most commonly used today is usually made of white plastic and is a pluggable (solderless) breadboard. It was designed by Ronald J. Portugal of EI Instruments Inc. in 1971.

1.2.13.1.1 Alternatives

Alternative methods to create prototypes are point-to-point construction (reminiscent of the original wooden breadboards), wire wrap, wiring pencil, and boards like the stripboard. Complicated systems, such as modern computers comprising millions of transistors, diodes, and resistors, do not lend themselves to prototyping using breadboards, as their complex designs can be difficult to lay out and debug on a breadboard.

Modern circuit designs are generally developed using a schematic capture and simulation system, and tested in software simulation before the first prototype circuits are built on a printed circuit board. Integrated circuit designs are a more extreme version of the same process: since producing prototype silicon is costly, extensive software simulations are performed before fabricating the first prototypes. However, prototyping techniques are still used for some applications such as RF circuits, or where software models of components are inexact or incomplete.

You could also use a square grid of pairs of holes where one hole per pair connects to its row and the other connects to its column. This same shape can be in a circle with rows and columns each spiraling opposite clockwise/counterclockwise.

1.2.13.2 Solderless breadboard

1.2.13.2.1 Typical specifications

A modern solderless breadboard consists of a perforated block of plastic with numerous tin plated phosphor bronze or nickel silver alloy spring clips under the perforations. The clips are often called *tie points* or *contact points*. The number of tie points is often given in the specification of the breadboard.

The spacing between the clips (lead pitch) is typically 0.1 in (2.54 mm). Integrated circuits (ICs) in dual in-line packages (DIPs) can be inserted to straddle the centerline of the block. Interconnecting wires and the leads of discrete components (such as

capacitors, resistors, and inductors) can be inserted into the remaining free holes to complete the circuit. Where ICs are not used, discrete components and connecting wires may use any of the holes. Typically the spring clips are rated for 1 ampere at 5 volts and 0.333 amperes at 15 volts (5 watts).

1.2.13.2.2 Bus and terminal strips

Solderless breadboards are available from several different manufacturers, but most share a similar layout. The layout of a typical solderless breadboard is made up from two types of areas, called strips. Strips consist of interconnected electrical terminals.

Terminal strips

> **The main areas, to hold most of the electronic components.**
>
> **In the middle of a terminal strip of a breadboard, one typically finds a notch running in parallel to the long side. The notch is to mark the centerline of the terminal strip and provides limited airflow (cooling) to DIP ICs straddling the centerline. The clips on the right and left of the notch are each connected in a radial way; typically five clips (i.e., beneath five holes) in a row on each side of the notch are electrically connected. The five clip columns on the left of the notch are often marked as A, B, C, D, and E, while the ones on the right are marked F, G, H, I and J. When a "skinny" dual in-line pin package (DIP) integrated circuit (such as a typical DIP-14 or DIP-16, which have a 0.3-inch (7.6 mm) separation between the pin rows) is plugged into a breadboard, the pins of one side of the chip are supposed to go into column E while the pins of the other side go into column F on the other side of the notch.**

Bus strips

> **To provide power to the electronic components.**
>
> **A bus strip usually contains two columns: one for ground and one for a supply voltage. However, some breadboards only provide a single-column power distributions bus strip on each long side. Typically the column intended for a supply voltage is marked in red, while the column for ground is marked in blue or black. Some manufacturers connect all terminals in a column. Others just connect groups of, for example, 25 consecutive terminals in a column. The latter design provides a circuit designer with some more control over crosstalk (inductively coupled noise) on the power supply bus. Often the groups in a bus strip are indicated by gaps in the color marking.**

Bus strips typically run down one or both sides of a terminal strip or between terminal strips. On large breadboards additional bus strips can often be found on the top and bottom of terminal strips.

Some manufacturers provide separate bus and terminal strips. Others just provide breadboard blocks which contain both in one block. Often breadboard strips or blocks of one brand can be clipped together to make a larger breadboard.

In a more robust variant, one or more breadboard strips are mounted on a sheet of metal. Typically, that backing sheet also holds a number of binding posts. These posts provide a clean way to connect an external power supply. This type of breadboard may be slightly easier to handle. Several images in this article show such solderless breadboards.

1.2.13.2.2.1 Diagram

A "full size" terminal breadboard strip typically consists of around 56 to 65 rows of connectors, each row containing the above-mentioned two sets of connected clips (A to E and F to J). Together with bus strips on each side this makes up a typical 784 to 910 tie point solderless breadboard. "Small size" strips typically come with around 30 rows. Miniature solderless breadboards as small as 17 rows (no bus strips, 170 tie points) can be found, but these are only suitable for small and simple designs.

1.2.13.2.3 Jump wires

Jump wires (also called jumper wires) for solderless breadboarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Ready-to-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm^2) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped $^3⁄_{16}$ to $^5⁄_{16}$ in (4.8 to 7.9 mm). Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.

Differently colored wires and color-coding discipline are often adhered to for consistency. However, the number of available colors is typically far fewer than the number of signal types or paths. Typically, a few wire colors are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used where convenient. Some ready-to-use jump wire sets use the color to indicate the length of the wires, but these sets do not allow a meaningful color-coding schema.

1.2.13.2.4 Advanced solderless breadboards

Some manufacturers provide high-end versions of solderless breadboards. These are typically high-quality breadboard modules mounted on a flat casing. The casing contains additional equipment for breadboarding, such as a power supply, one or more signal generators, serial interfaces, LED or LCD display modules, and logic probes.

Solderless breadboard modules can also be found mounted on devices like microcontroller evaluation boards. They provide an easy way to add additional periphery circuits to the evaluation board.

1.2.13.2.5 High frequencies and dead bugs

For high-frequency development, a metal breadboard affords a desirable solderable ground plane, often an unetched piece of printed circuit board; integrated circuits are sometimes stuck upside down to the breadboard and soldered to directly, a technique sometimes called "dead bug" construction because of its appearance. Examples of dead bug with ground plane construction are illustrated in a Linear Technologies application note. For other uses of this technique see dead bugs.

1.2.13.2.6 Limitations

Due to relatively large stray capacitance compared to a properly laid out PCB (approx 2pF between adjacent contact columns), high inductance of some connections and a relatively high and not very reproducible contact resistance, solderless breadboards are limited to operation at relatively low frequencies, usually less than 10 MHz, depending on the nature of the circuit. The relatively high contact resistance can already be a

problem for some DC and very low frequency circuits. Solderless breadboards are further limited by their voltage and current ratings.

Solderless breadboards usually cannot accommodate surface-mount technology devices (SMD) or components with grid spacing other than 0.1 in (2.54 mm). Further, they cannot accommodate components with multiple rows of connectors if these connectors don't match the dual in-line layout—it is impossible to provide the correct electrical connectivity. Sometimes small PCB adapters called "breakout adapters" can be used to fit the component to the board. Such adapters carry one or more components and have 0.1 in (2.54 mm) spaced male connector pins in a single in-line or dual in-line layout, for insertion into a solderless breadboard. Larger components are usually plugged into a socket on the adapter, while smaller components (e.g., SMD resistors) are usually soldered directly onto the adapter. The adapter is then plugged into the breadboard via the 0.1 in (2.54 mm) connectors. However, the need to solder the components onto the adapter negates some of the advantage of using a solderless breadboard.

Very complex circuits can become unmanageable on a solderless breadboard due to the large amount of wiring required. The very convenience of easy plugging and unplugging of connections also makes it too easy to accidentally disturb a connection, and the system becomes unreliable. It is possible to prototype systems with thousands of connecting points, but great care must be taken in careful assembly, and such a system becomes unreliable as contact resistance develops over time. At some point, very complex systems must be implemented in a more reliable interconnection technology, to have a likelihood of working over a usable time period.

CHAPTER 2:Introduction to security

We start our description of security in distributed systems by taking a look at some general security issues. First, it is necessary to define what a secure system is.

We distinguish security policies from security mechanisms, and take a look at the Globus wide-area system for which a security policy has been explicitly formulated.

Our second concern is to consider some general design issues for secure systems.

Finally, we briefly discuss some cryptographic algorithms, which play a key role in the design of security protocols.

Security Threats, Policies, and Mechanisms Security in computer systems is strongly related to the notion of dependability. Informally, a dependable computer system is one that we justifiably trust to deliver its services (Laprie, 1995).

Confidentiality refers to the property of a computer system whereby its information is disclosed only to authorized parties.

Integrity is the characteristic that alterations to a system's assets can be made only in an authorized way.

In other words, improper alterations in a secure computer system should be detectable and recoverable.

Major assets of any computer system are its hardware, software, and data. Another way of looking at security in computer systems is that we attempt to protect the services and data it offers against security threats.

There are four types of security threats to consider (Pfleeger, 1997):

1. Interception
2. Interruption
3. Modification
4. Fabrication

Interception refers to the situation that an unauthorized party has gained access to a service or data.

A typical example of interception is where communication between two parties has been overheard by someone else.

Interception also happens when data are illegally copied, for example, after breaking into a person's private directory in a file system.

An example of interruption is when a file is corrupted or lost.

In general, interruption refers to the situation in which services or data become unavailable, unusable, destroyed, and so on.

In this sense, denial of service attacks by which someone maliciously attempts to make a service inaccessible to other parties is a security threat that classifies as interruption.

Modifications involve unauthorized changing of data or tampering with a service so that it no longer adheres to its original specifications.

Examples of modifications include intercepting and subsequently changing transmitted data, tampering with database entries, and changing a program so that it secretly logs the activities of its user.

Fabrication refers to the situation in which additional data or activity are generated that would normally not exist.

For example, an intruder may attempt to add an entry into a password file or database.

Likewise, it is sometimes possible to break into a system by replaying previously sent messages.

We shall come across such examples later in this chapter.

Note that interruption, modification, and fabrication can each be seen as a form of data falsification.

Simply stating that a system should be able to protect itself against all possible security threats is not the way to actually build a secure system.

What is first needed is a description of security requirements, that is, a security policy.

A security policy describes precisely which actions the entities in a system are allowed to take and which ones are prohibited.

Entities include users, services, data, machines, and so on.

Once a security policy has been laid down, it becomes possible to concentrate on the security mechanisms by which a policy can be enforced.

Important security mechanisms are:

1. Encryption

2. Authentication

3. Authorization

4. Auditing

Encryption is fundamental to computer security.

Encryption transforms data into something an attacker cannot understand.

In other words, encryption provides a means to implement confidentiality.

In addition, encryption allows us to check whether data have been modified.

It thus also provides support for integrity checks.

Authentication is used to verify the claimed identity of a user, client, server, and so on. In the case of clients, the basic premise is that before a service will do work for a client, the service must learn the client's identity.

Typically, users are authenticated by means of passwords, but there are many other ways to authenticate clients.

After a client has been authenticated, it is necessary to check whether that client is authorized to perform the action requested.

Access to records in a medical database is a typical example.

Depending on who accesses the database, permission may be granted to read records, to modify certain fields in a record, or to add or remove a record.

Auditing tools are used to trace which clients accessed what, and which way.

Although auditing does not really provide any protection against security threats, audit logs can be extremely useful for the analysis of a security breach, and subsequently taking measures against intruders.

For this reason, attackers are generally keen not to leave any traces that could eventually lead to exposing their identity.

In this sense, logging accesses makes attacking sometimes a riskier business.

Example: The Globus Security Architecture

The notion of security policy and the role that security mechanisms play in distributed systems for enforcing such policies is often best explained by taking a look at a concrete example.

Consider the security policy defined for the Globus wide-area system (Chervenak et al., 2000).

Globus is a system supporting largescale distributed computations in which many hosts, files, and other resources are simultaneously used for doing a computation.

Such environments are also referred to as computational grids (Foster and Kesselman, 1998).

Resources in these grids are often located in different administrative domains that may be located in different parts of the world.

Because users and resources are vast in number and widely spread across different administrative domains, security is essential.

To devise and properly use security mechanisms, it is necessary to understand what exactly needs to be protected, and what the assumptions are with respect to security.

Simplifying matters somewhat, the security policy for Globus entails the following eight statements, which we explain below (Foster et al., 1998):

1. The environment consists of multiple administrative domains.

2. Local operations (i.e., operations that are carried out only within a single domain) are subject to a local domain security policy only.

3. Global operations (i.e., operations involving several domains) require the initiator to

be known in each domain where the operation is carried out.

4. Operations between entities in different domains require mutual authentication.

5. Global authentication replaces local authentication.

6. Controlling access to resources is subject to local security only.

7. Users can delegate rights to processes.

8. A group of processes in the same domain can share credentials.

Globus assumes that the environment consists of multiple administrative domains, where each domain has its own local security policy.

It is assumed that local policies cannot be changed just because the domain participates in Globus, nor can the overall policy of Globus override local security decisions.

Consequently, security in Globus will restrict itself to operations that affect multiple domains.

Related to this issue is that Globus assumes that operations that are entirely local to a domain are subject only to that domain's security policy.

In other words, if an operation is initiated and carried out within a single domain, all security issues will be carried out using local security measures only.

Globus will not impose additional measures.

The Globus security policy states that requests for operations can be initiated either globally or locally.

The initiator, be it a user or process acting on behalf of a user, must be locally known within each domain where that operation is carried out.

For example, a user may have a global name that is mapped to domainspecific local names.

How exactly that mapping takes place is left to each domain.

An important policy statement is that operations between entities in different domains require mutual authentication.

This means, for example, that if a user in one domain makes use of a service from other domain, then the identity of the user will have to be verified.

Equally important is that the user will have to be assured that he is using a service he thinks he is using.

We return to authentication, extensively, later in this chapter.

The above two policy issues are combined into the following security requirement.

If the identity of a user has been verified, and that user is also known locally in a domain, then he can act as being authenticated for that local domain.

This means that Globus requires that its system wide authentication measures are sufficient to consider that a user has already been authenticated for a remote domain (where that user is known) when accessing resources in that domain.

Additional authentication by that domain should not be necessary.

Once a user (or process acting on behalf of a user) has been authenticated, it is still necessary to verify the exact access rights with respect to resources.

For example, a user wanting to modify a file will first have to be authenticated, after which it can be checked whether or not that user is actually permitted to modify the file.

The Globus security policy states that such access control decisions are made entirely local within the domain where the accessed resource is located.

To explain the seventh statement, consider a mobile agent in Globus that carries out a task by initiating several operations in different domains, one after another.

Such an agent may take a long time to complete its task.

To avoid communication with the user on whose behalf the agent is acting, Globus requires that processes can be delegated a subset of the user's rights.

Consequently, by authenticating an agent and subsequently checking its rights, Globus should be able to allow an agent to initiate an operation without having to contact the agent's owner.

As a final policy statement, Globus requires that groups of processes running with a single domain and acting on behalf of the same user may share a single set of credentials.

As will be explained below, credentials are needed for authentication.

This statement essentially opens the road to scalable solutions for authentication by not demanding that each process carries its own unique set of credentials.

The Globus security policy allows its designers to concentrate on developing an overall solution for security.

By assuming that each domain enforces its own security policy, Globus concentrates only on security threats involving multiple domains.

In particular, the security policy indicates that the important design issues are the representation of a user in a remote domain, and the allocation of resources from a remote domain to a user or his representative.

What Globus therefore primarily needs, are mechanisms for cross-domain authentication, and making a user known in remote domains.

For this purpose, two types of representatives are introduced.

A user proxy is a process that is given permission to act on behalf of a user for a limited period of time.

Resources are represented by resource proxies.

A resource proxy is a process running within a specific domain that is used to translate global operations on a resource into local operations that comply with that particular domain's security policy.

For example, a user proxy will typically communicate with a resource proxy when access to that resource is required.

The Globus security architecture essentially consists of entities such as users, user proxies, resource proxies, and general processes.

These entities are located in domains and interact with each other. In particular, the security architecture defines four different protocols, as shown in Fig.

The first protocol describes precisely how a user can create a user proxy and delegate rights to that proxy.

In particular, in order to let the user proxy act on behalf of its user, the user gives the proxy an appropriate set of credentials.

The second protocol specifies how a user proxy can request the allocation of a resource in a remote domain.

In essence, the protocol tells a resource proxy to create a process in the remote domain after mutual authentication has taken place.

That process represents the user (just as the user proxy did), but operates in the same domain as the requested resource.

The process is given access to the resource subject to the access control decisions local to that domain.

A process created in a remote domain may initiate additional computations in other domains.

Consequently, a protocol is needed to allocate resources in a remote domain as requested by a process other than a user proxy.

In Globus, this type of allocation is done via the user proxy, by letting a process have its associated user proxy request the allocation of resources, essentially following the second protocol.

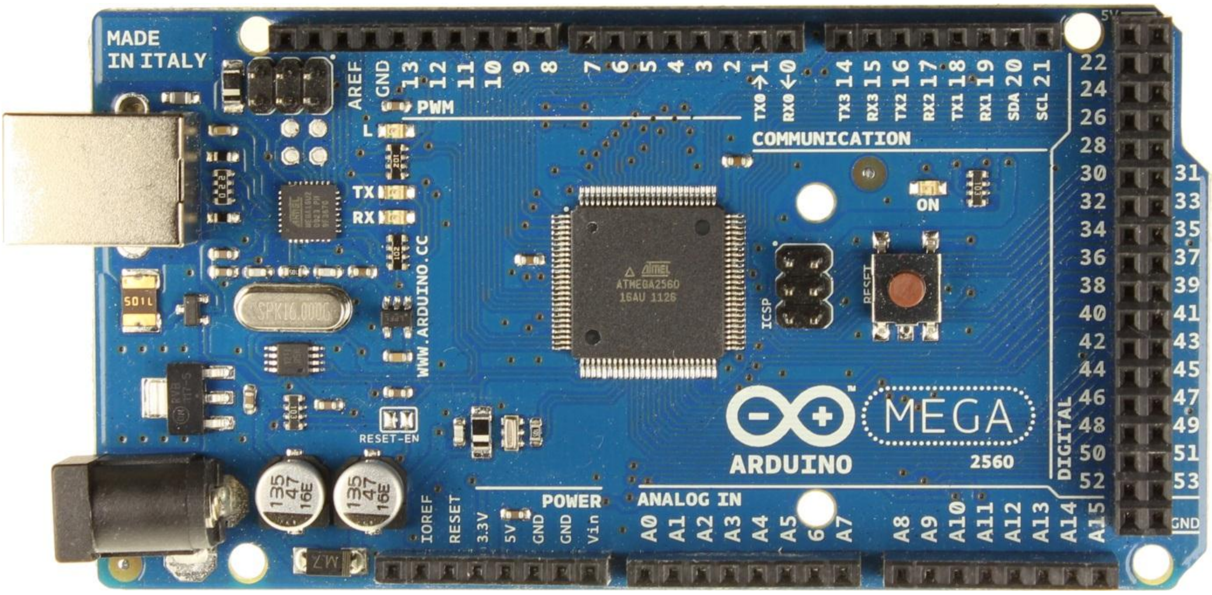
The fourth and last protocol in the Globus security architecture is the way a user can make himself known in a domain.

Assuming that a user has an account in a domain, what needs to be established is that the systemwide credentials as held by a user proxy are automatically converted to credentials that are recognized by the specific domain.

The protocol prescribes how the mapping between the global .

CHAPTER 3: Components of the project

3.1 Arduino Mega 2560



PRODUCT DESCRIPTION

Original Arduino Italy Product (shipped free to anywhere in Egypt)

Description

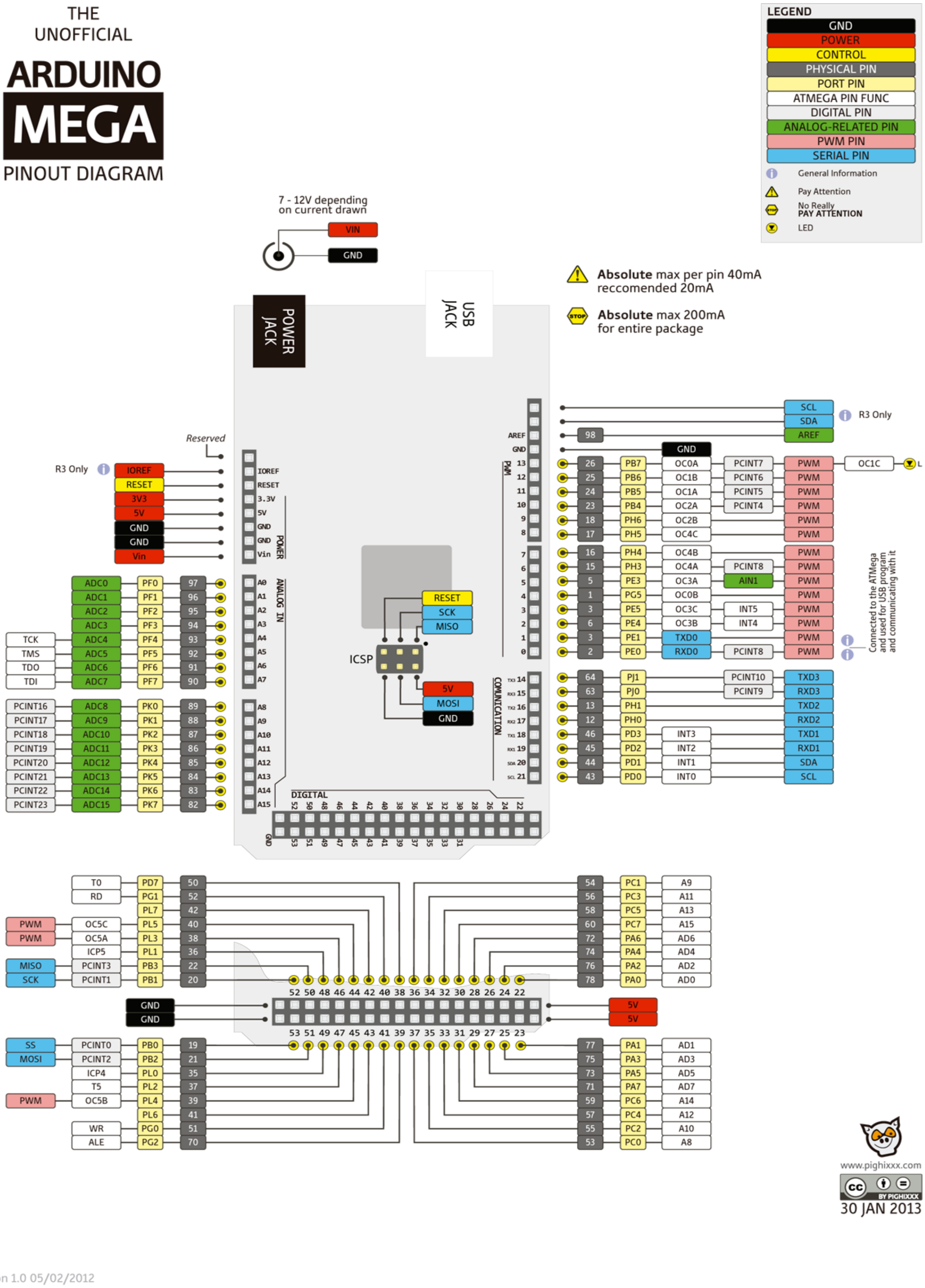
The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Specifications

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader

SRAM	**8 KB**
EEPROM	**4 KB**
Clock Speed	**16 MHz**



The following related products will be added to your cart automatically with this purchase:

3.2 9V Battery Adapter for the Arduino

PRODUCT DESCRIPTION

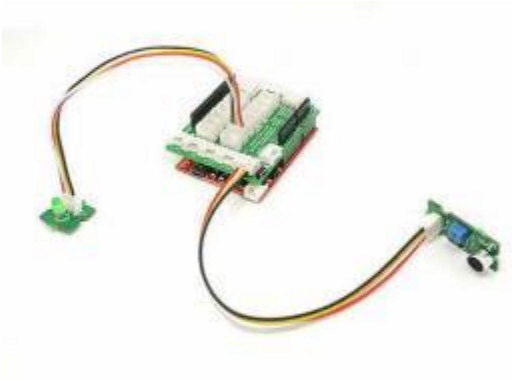
Use this cable to battery-power any device that needs 9V and has an on-board barrel jack - it works great for Arduinos, development boards, evaluation boards, and more!

3.3 High Quality Arduino USB programming Cable


PRODUCT DESCRIPTION

High quality, high speed USB2.0 cable It is compatible with all devices USB A male to USB B male. Used for arduino boards programming.

3.4 Sound Sensor (adjustable)



PRODUCT DESCRIPTION

The Sound sensor module can detect sound strength (intensity) with high sensitivity. It can be used to detect the sound strength of the environment. The value of output can be adjusted by the potentiometer.

Features

- **Wide supply voltage range: 4V-12V**
- **Low quiescent current drain: 4mA**
- **High sensitivity**
- **Gain adjustable**

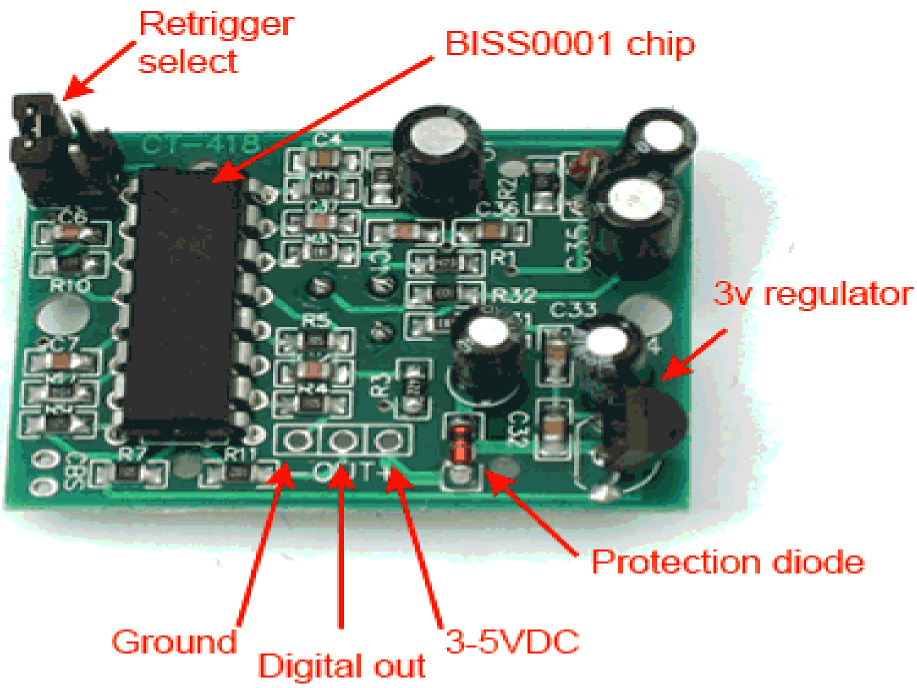
Applications Ideas

- **Simple microphone**
- **Sound detection**

3.5 PIR Motion sensor module (Adjustable Range)

PRODUCT DESCRIPTION

PIR "Passive Infrared"sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.



How to Use PIR Sensor

The PIR sensor have 3 pins:

Power: **3-5 VDC**

Output: **High (when motion detected)**

Low (no motion detected)

Ground: **connected to ground**

The output pin will go "high" to 3.3V and light up the LED when motion detected.

Specifications:

Input Voltage: **DC 3-5V**

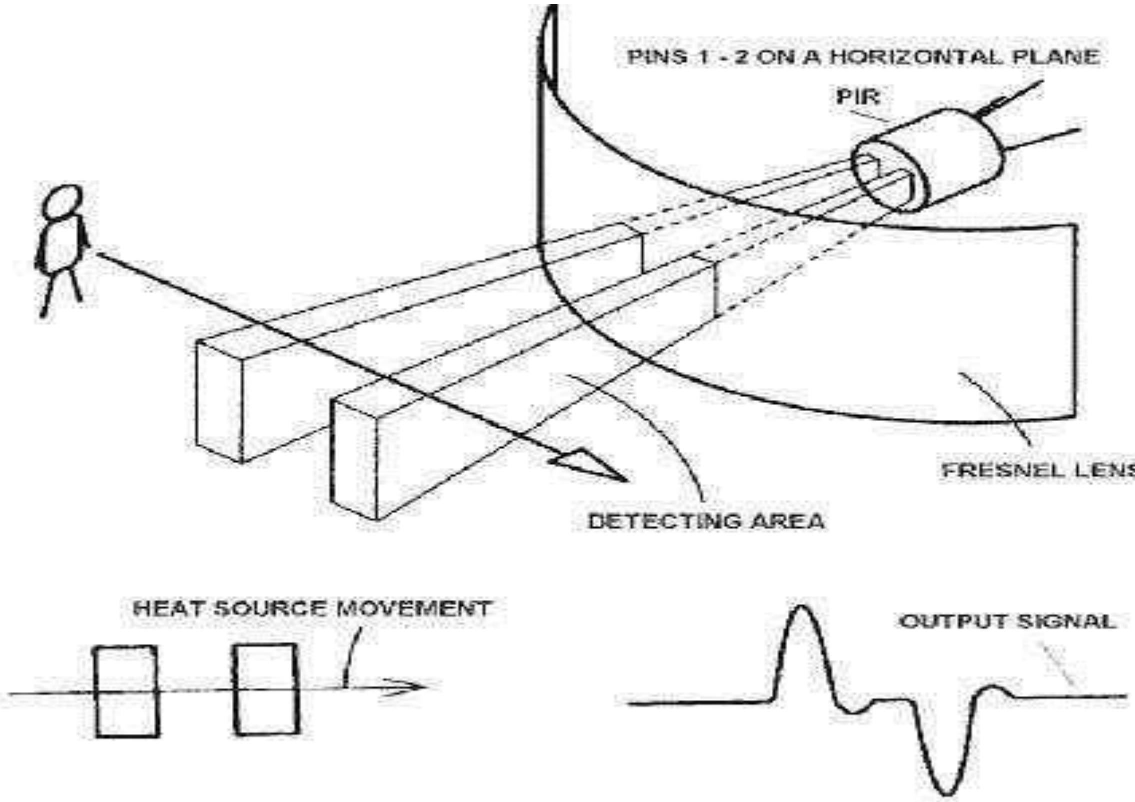
Output signal: **3.3V (motion detected-output high)**

Detection Angle: **110 degree**

Detection Angle: **max 7 m**

1 How Does The PIR Sensor Work

The PIR sensor itself has two slots each one of them is made of a special material that is sensitive to infrared. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor. This causes a positive differential change between the two halfes. These change pulses are what is detected.

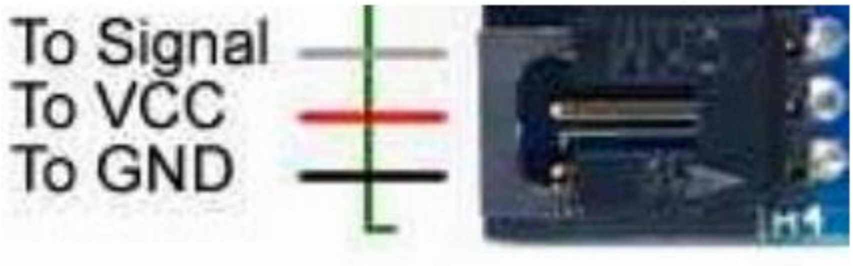


This PIR Sensor Switch Module detects the Infrared Rays released by human body motion within the detection area 7 Meters.

For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by house pets.

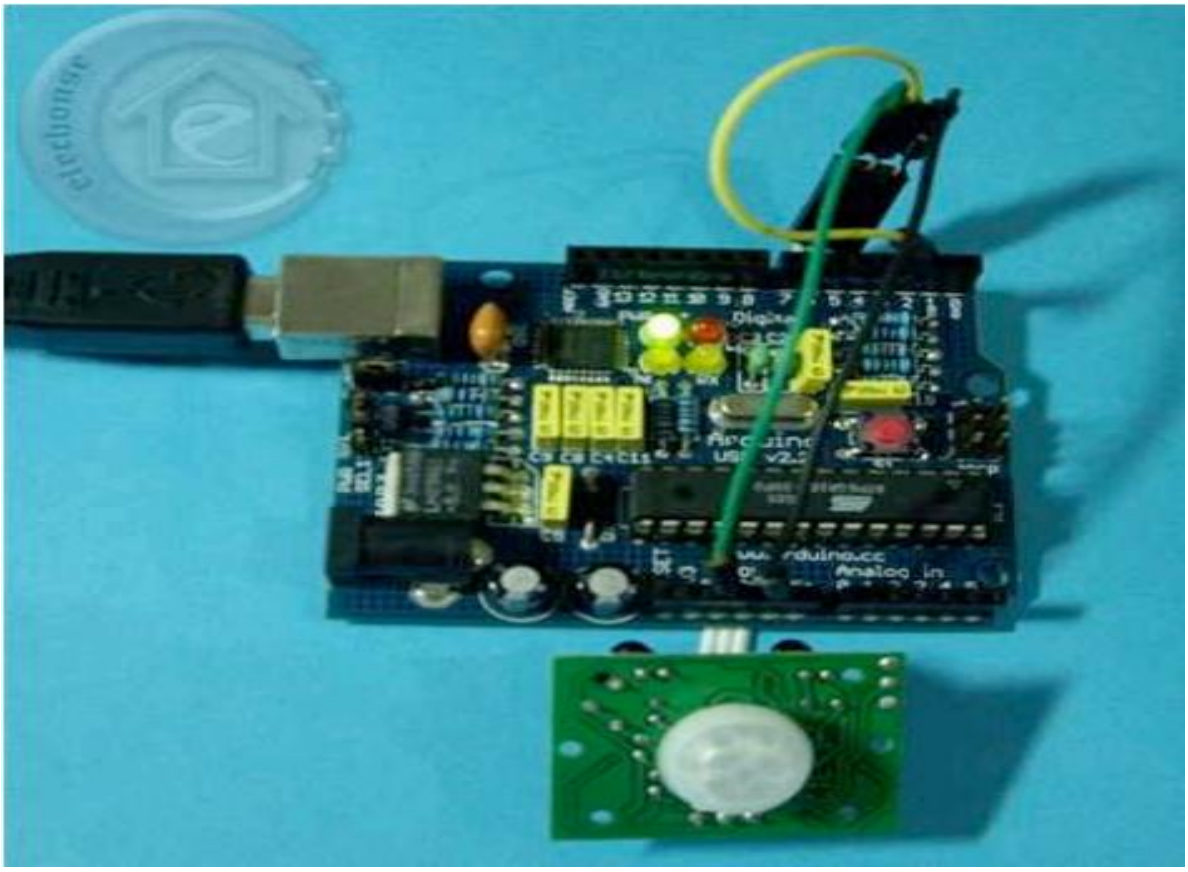
Technical Data

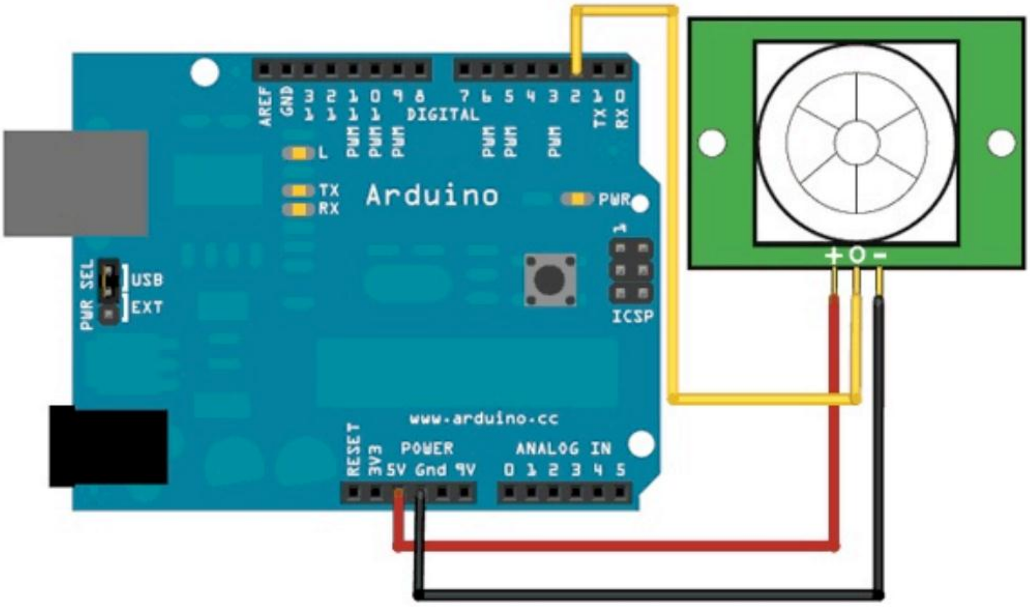
Each PIR module has buckled port with 3 pin connection VCC, GND and Output Signal, as shown in the figure below:



Output (applied to output signal pin): Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). The signal duration can be adjusted from 0.3s-18s by the yellow potentiometer.
Input voltage (applied to VCC pin): 3.3V - 5V

Sensitivity range: up to 7 meters 110 degrees x 70 degrees detection range.
Connecting to Microcontrollers and Arduino
Connecting PIR sensors to a microcontroller is really simple. The PIR acts as a digital output so all you need to do is listen for the signal pin to flip high (detected) or low (not detected) (see figure below).





Software (code for Arduino) **the corresponding Arduino code is :**

```
int ledPin = 13;

int switchPin = 2;

int value = 0;

void setup() {
```

```
pinMode(ledPin, OUTPUT);
pinMode(switchPin, INPUT);
}
void loop() {
value = digitalRead(switchPin);
if (HIGH == value) {
digitalWrite(ledPin, HIGH);
} else {
digitalWrite(ledPin, LOW);
}
}
```

3.6 LM35DZ Temperature Sensor (Precision Centigrade)

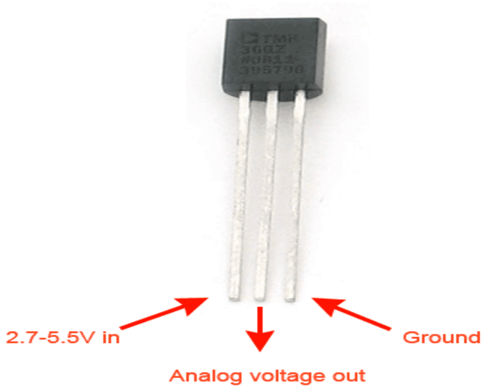
PRODUCT DESCRIPTION

The LM35 series are precision IC temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling.

The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full ?55°C to +150°C temperature range.

Features

- **Calibrated directly in Celsius**
- **Linear + 10 mV/°C scale factor**
- **0.5°C ensured accuracy (at +25°C)**



Applications

- **Sensing & Instrumentation**
- **Consumer Electronics**
- **Automation & Process Control**
- **System Monitoring**

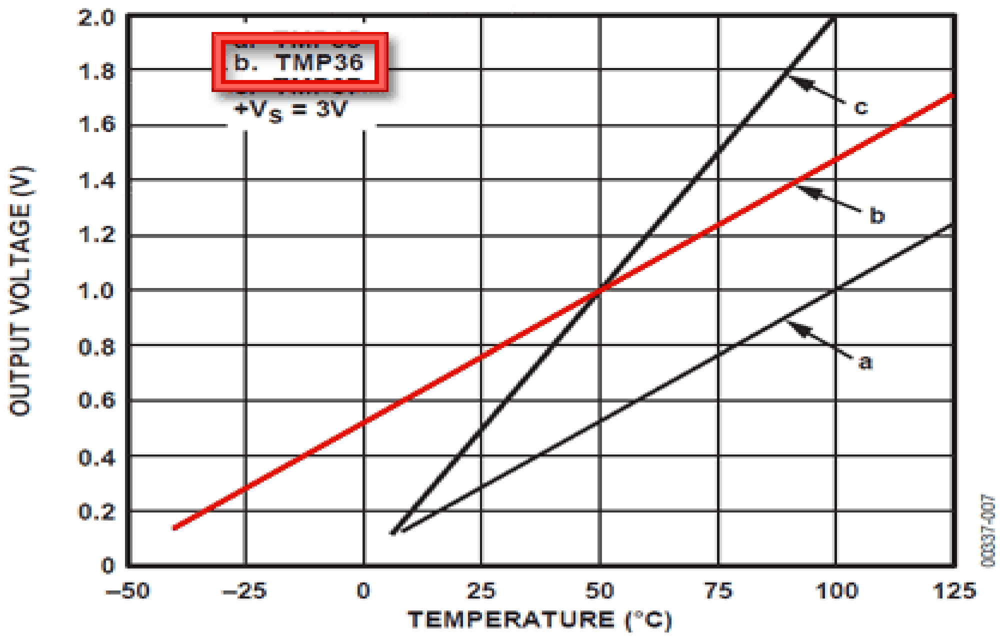


Figure 6. Output Voltage vs. Temperature

Arduino LM35 Sensor

My son Paul and I recently finished a project using the Arduino Diecimila microcontroller in conjunction with the Processing open source programming environment to monitor temperature.

The project contains 3 parts:

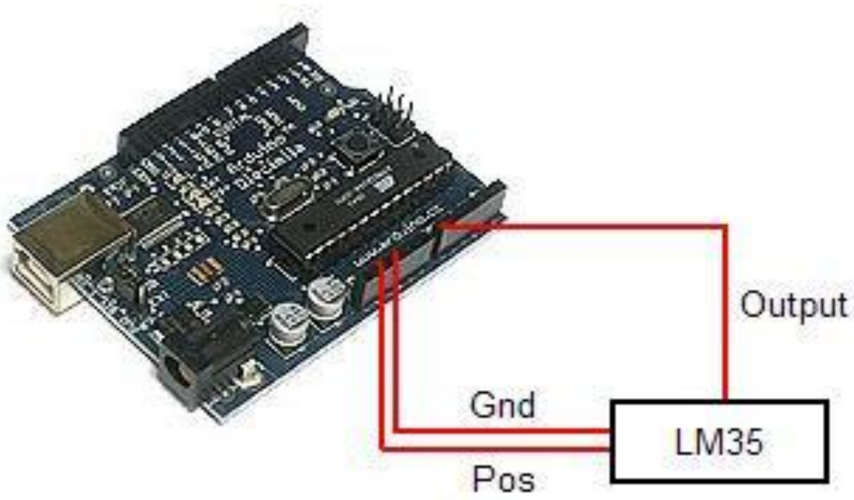
1. The Arduino board with sensor circuit.

2. The Arduino program.

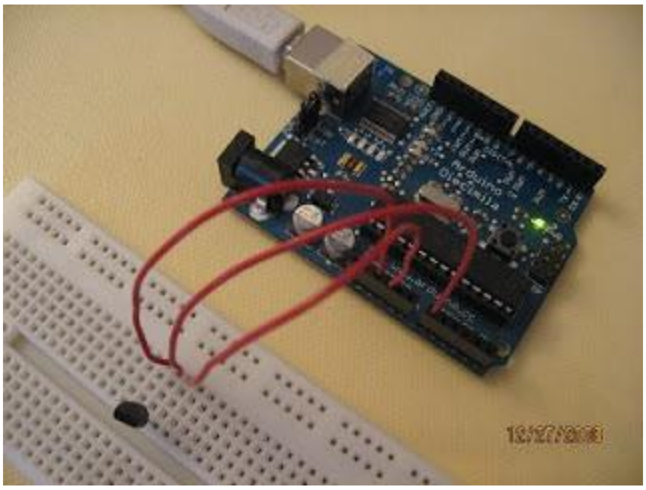
3. The Processing program.

The Arduino Board with Sensor Circuit

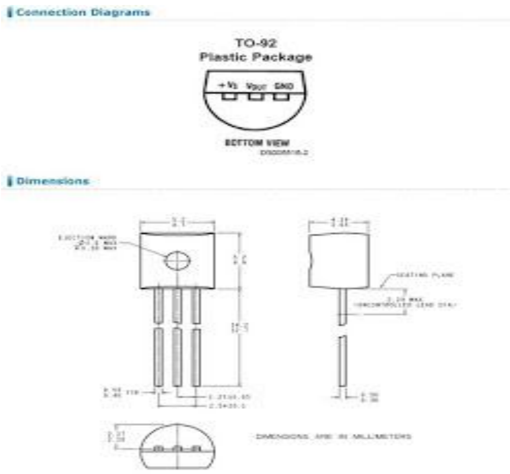
The Arduino circuit board is connected to the LM35 Celsius Temperature sensor. Here is a picture of the project circuit with illustrated wires connected to the temperature sensor.



We used the on board power source (5v and Gnd) to power the LM35 and analog pin 0 (zero) to read the analog output from the sensor. Here's a picture of the circuit wired on a breadboard.



The LM35 temperature sensor's pin-out and package information is as follows:



The Arduino Program

The open-source Arduino environment allows us to write code and load it onto the Arduino board's memory. The development environment is written in Java and based on Processing, avr-gcc, and other open source software.
The Arduino code loops every second to read output from the LM35, converting the analog output into Celsius and sending the data to the computer via a serial communication connection (USB).

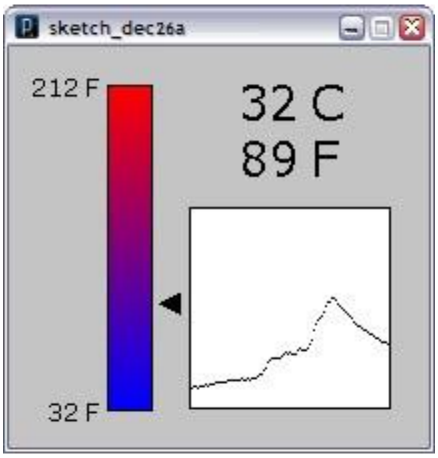
Here's the code used to run the Arduino board:

```
//declare variables
float tempC;
int tempPin = 0;

void setup()
{
Serial.begin(9600); //opens serial port, sets data rate to 9600 bps
}

void loop()
{
tempC = analogRead(tempPin);           //read the value from the
sensor
tempC = (5.0 * tempC * 100.0)/1024.0;  //convert the analog data to
temperature
Serial.print((byte)tempC);             //send the data to the
computer
delay(1000);                           //wait one second before
sending new data
}
```

The Processing Program

The software client portion of this project runs on a PC and is written in Processing. Processing is a language and development environment similar to Arduino and designed for creating visual effects programs. We used Processing to create a small client that can read the serial data from the Arduino board and display the temperature on a slider and in both Celsius and Fahrenheit. We also added a rolling 100 data point graph to display

historical temperature data. Here's a screen shot of the Processing application:



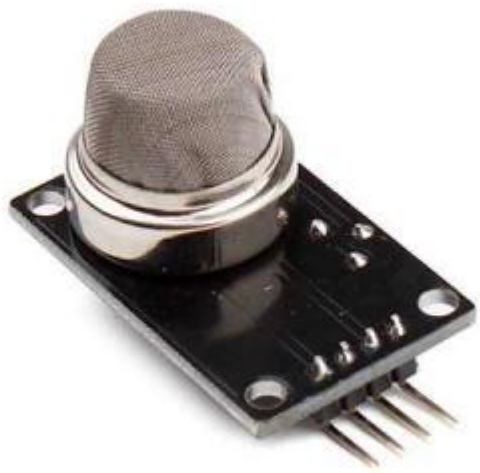
Here is the code used for the visual portion of the project:

```
//import Serial communication library
import processing.serial.*;

//init variables
Serial commPort;
float tempC;
float tempF;
int yDist;
PFont font12;
PFont font24;
float[] tempHistory = new float[100];

void setup()
{
  //setup fonts for use throughout the application
  font12 = loadFont("Verdana-12.vlw");
  font24 = loadFont("Verdana-24.vlw");

  //set the size of the window
  size(210, 200);
```

```
  //init serial communication port
  commPort = new Serial(this, "COM10", 9600);

  //fill tempHistory with default temps
  for(int index = 0; index<100; index++)
    tempHistory[index] = 0;
}

void draw()
{
  //get the temp from the serial port
  while (commPort.available() > 0)
  {
    tempC = commPort.read();

    //refresh the background to clear old data
    background(123);

    //draw the temp rectangle
    colorMode(RGB, 160);  //use color mode sized for fading
    stroke (0);
    rect (49,19,22,162);
    //fade red and blue within the rectangle
    for (int colorIndex = 0; colorIndex <= 160; colorIndex++)
    {
      stroke(160 - colorIndex, 0, colorIndex);
      line(50, colorIndex + 20, 70, colorIndex + 20);
    }

    //draw graph
    stroke(0);
    fill(255,255,255);
    rect(90,80,100,100);
    for (int index = 0; index<100; index++)
```

```
    {
      if(index == 99)
        tempHistory[index] = tempC;
      else
        tempHistory[index] = tempHistory[index + 1];
      point(90 + index, 180 - tempHistory[index]);
    }

    //write reference values
    fill(0,0,0);
    textFont(font12);
    textAlign(RIGHT);
    text("212 F", 45, 25);
    text("32 F", 45, 187);

    //draw triangle pointer
    yDist = int(160 - (160 * (tempC * 0.01)));
    stroke(0);
    triangle(75, yDist + 20, 85, yDist + 15, 85, yDist + 25);

    //write the temp in C and F
    fill(0,0,0);
    textFont(font24);
    textAlign(LEFT);
    text(str(int(tempC)) + " C", 115, 37);
    tempF = ((tempC*9)/5) + 32;
    text(str(int(tempF)) + " F", 115, 65);
  }
}
```

3.7 Smoke sensor Module MQ2 (Digital/Analog)



PRODUCT DESCRIPTION

This smoke sensor module with MQ2 Smoke sensor have both digital and analog outputs , you also can adjust the sensitivity with the potentioneter.

PIN Connection

<u>4 pins as follows:</u>

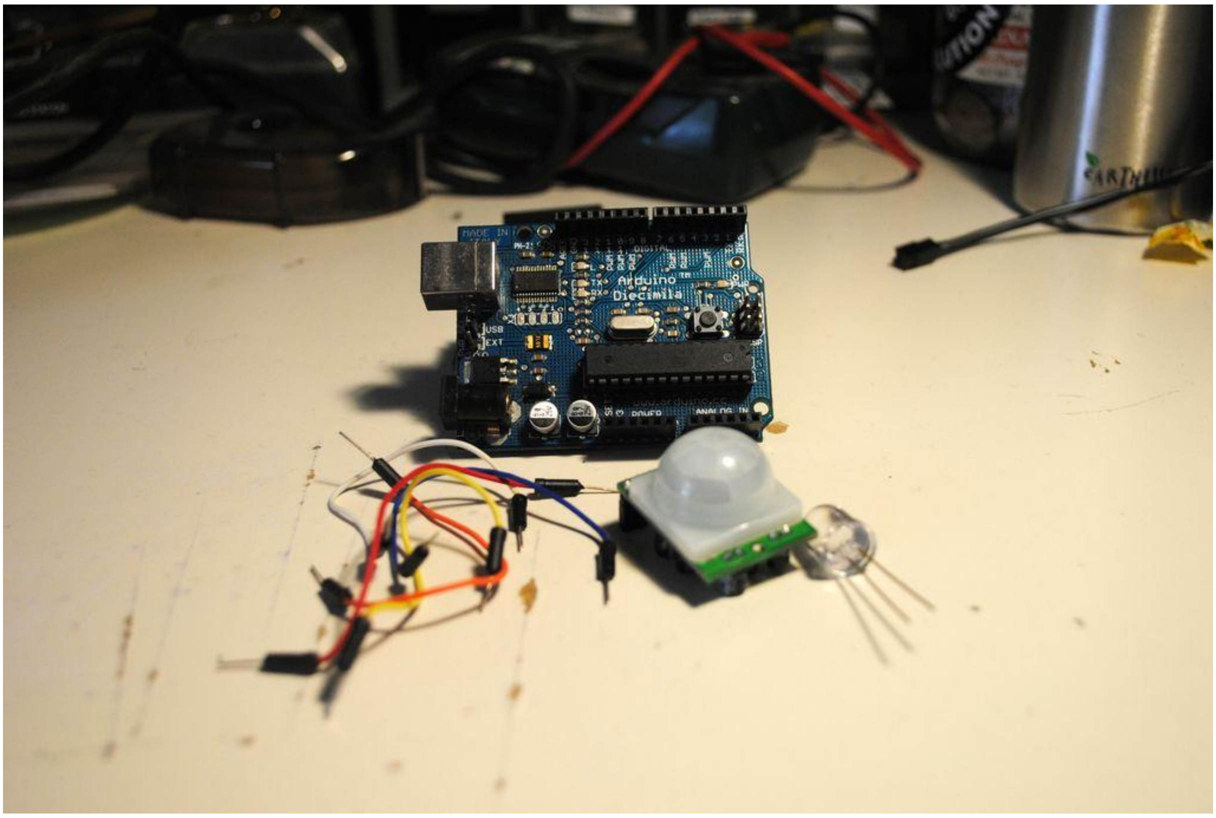
- **VCC   (5 V)**
- **GND (ground)**
- **Aout (Analog output)**
- **Dout (Digital output)**

When connect to arduino, use digital pins in digital mode (high or low) and use analog pins for analog mode (diffrenet value for different smoke intensity).

CHAPTER 4: Explain the project

4.1 PIR Sensors

4.1.1 Step 1: Supplies



For this instructable you will need the following:

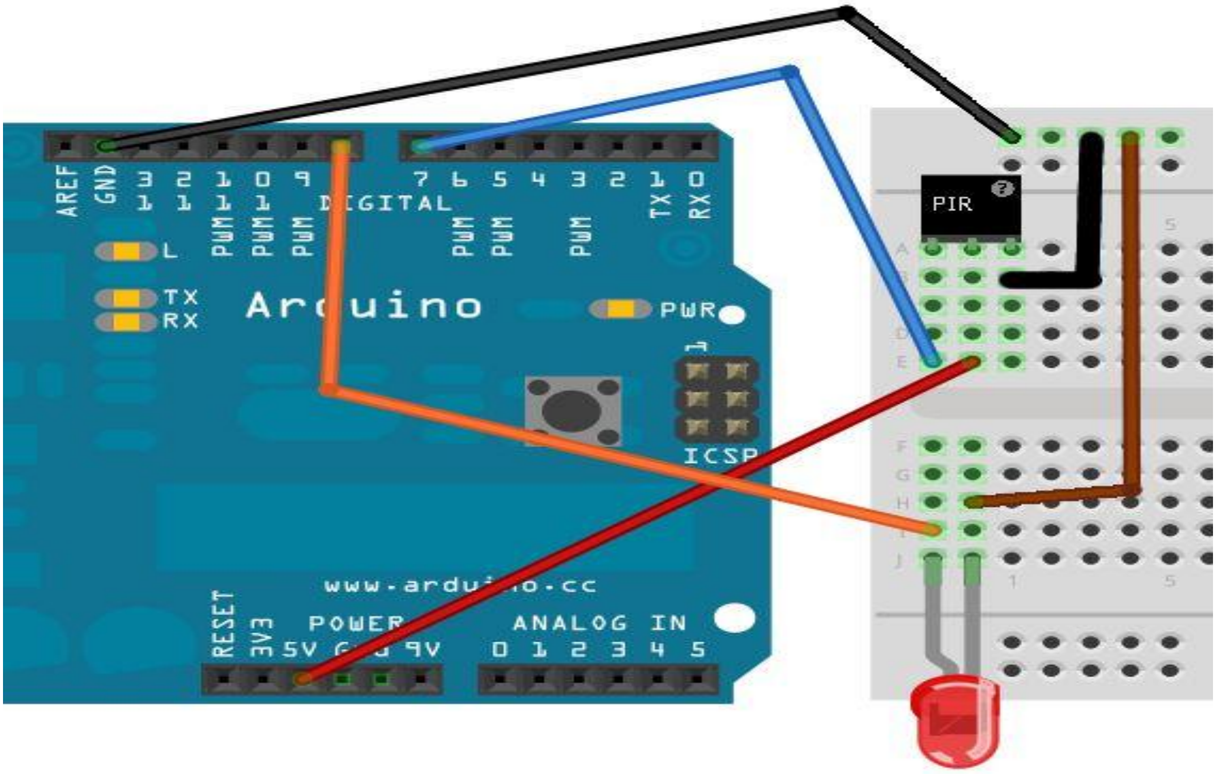
1 arduino (with protoshield to make life easy)

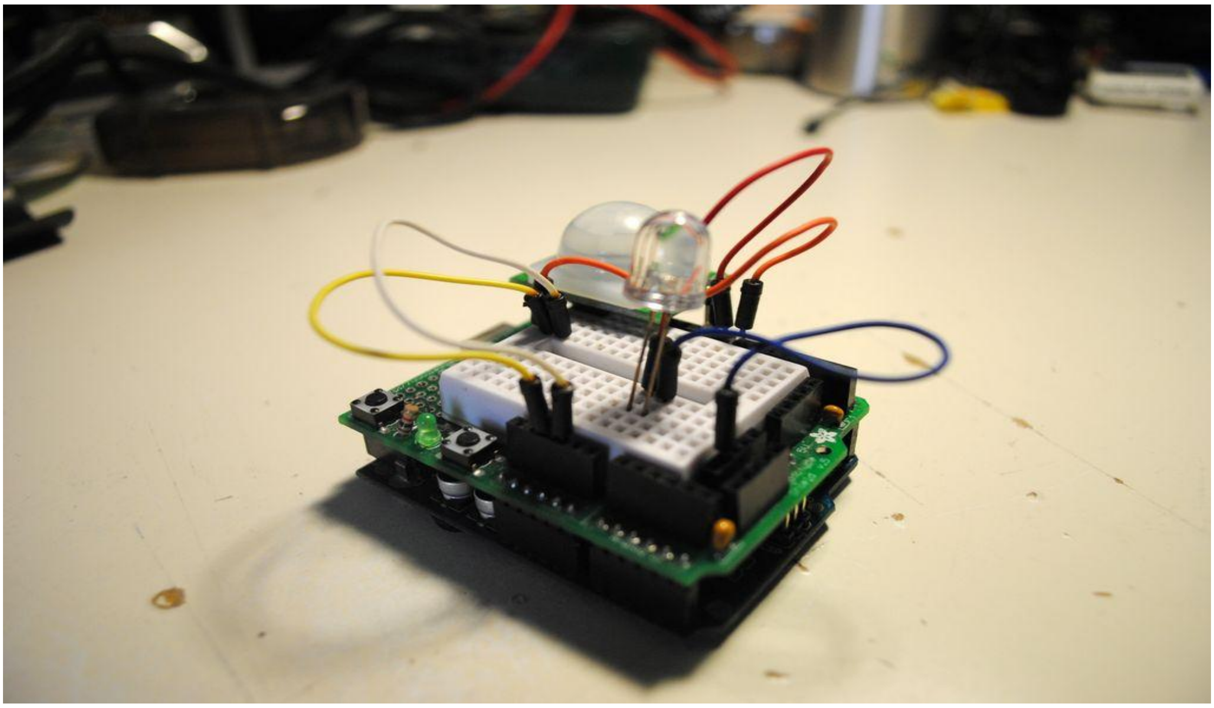
1 LED of any color

1 PIR sensor from Parallax (you can find these at most radio shacks)

Solid wire to hook it up

4.1.2 Step 2: Setup





The wiring is pretty simple, the PIR sensor has screen printed: + - out

Hook the + to 5v, - to ground and out to pin 7

The take the LED and put power to pin 8 and ground to ground.

If its confusing, take a look at the pictures!

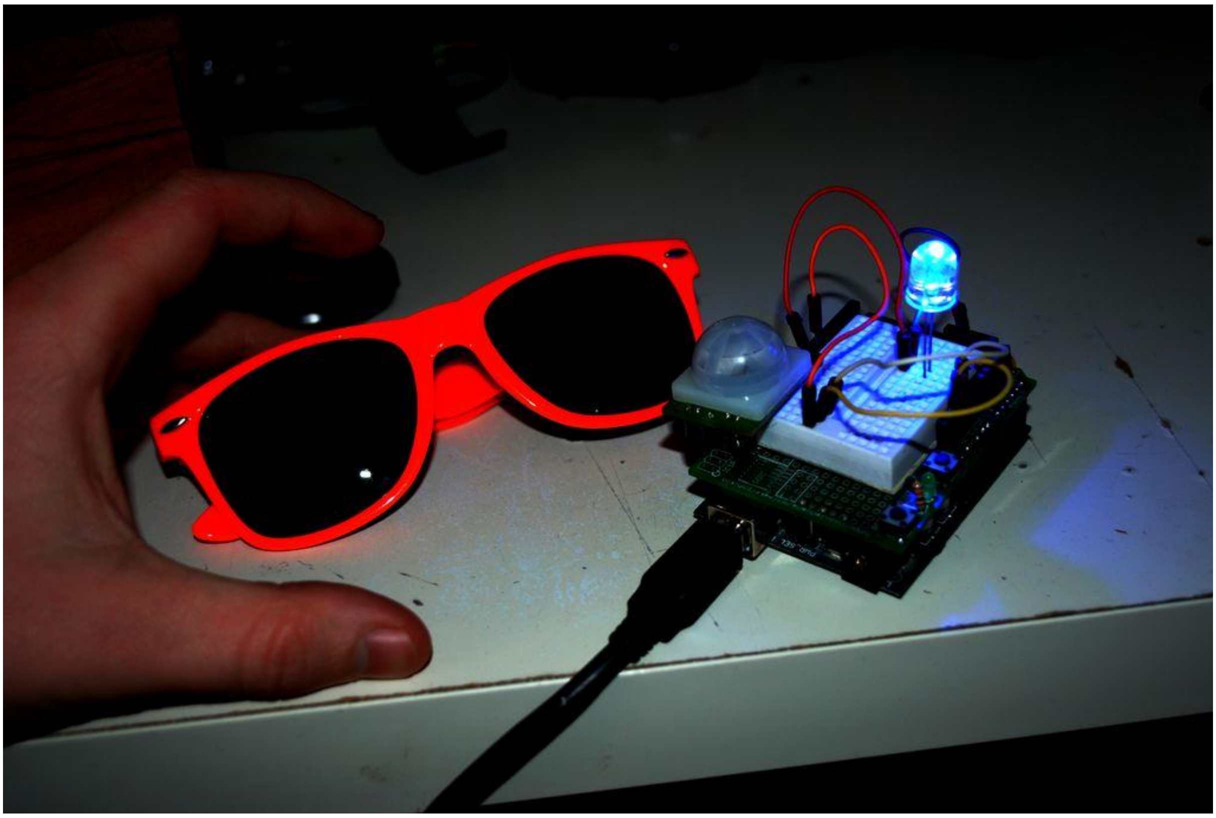
4.1.3 Step 3: Code

```
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;

int pirPin = 7;
int ledPin = 8;

void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);
Serial.print("calibrating sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(1000);
      }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(50);
  }
void loop(){

     if(digitalRead(pirPin) == HIGH){
       digitalWrite(ledPin, HIGH);
 output pin state
       if(lockLow){
```

```
        lockLow = false;
        Serial.println("---");
        Serial.print("motion detected at ");
        Serial.print(millis()/1000);
        Serial.println(" sec");
        delay(50);
        }
        takeLowTime = true;
      }

    if(digitalRead(pirPin) == LOW){
      digitalWrite(ledPin, LOW);
      if(takeLowTime){
       lowIn = millis();
       takeLowTime = false;
       }
      if(!lockLow && millis() - lowIn > pause){
          lockLow = true;
          Serial.print("motion ended at ");
          Serial.print((millis() - pause)/1000);
          Serial.println(" sec");
          delay(50);
          }
      }
  }
```

You can see from the code, the sensor first calibrates itself and then watches for movement. When it detects movement, the blue light goes on. You can watch the serial monitor to see how long the movement lasts.

4.1.4 Step 4: Further Projectse



After these steps you have a very simple motion detector. From here you can use the PIR sensor to trigger events (like a siren or a text message that someone is in your room).

I just chose to use it to protect my stunnaz from blue light fearing monsters. It seems to work so far....