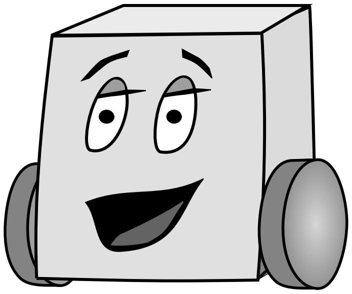


E11: Autonomous Vehicles

Fall 2010

Harris & Lape with Keeter & Ong

Lecture 2: Mudduino

An autonomous vehicle requires a controller to read and interpret sensors and issue commands to the motors and other actuators. In this class, you will build and use a Mudduino microcontroller board. This lecture describes the contents and capabilities of the board.

1. TERMINOLOGY

Let us begin with a few terms to describe electricity and electrical systems.

- **Current**: rate of flow of electric charge: dq/dt
- **Voltage**: electrical force that would drive an electric current between two points

A helpful analogy is that electricity flowing in a circuit is like water flowing in a pipe. Suppose one end of the pipe is elevated, causing water to flow. The difference in height between the ends is analogous to the voltage. The amount of water flowing through the pipe (e.g. gallons/sec) is analogous to the current; we even call it current when the pipe is a river! If the pipe were lying flat, there would be no reason for the water to flow from one end to the other, so the current becomes zero. In the same way, when no voltage is applied to a circuit, no current flows. Increasing the height of the end of the pipe or the voltage applied to the circuit causes more current to flow.

Electrical systems can be classified as analog or digital. Analog systems use a continuous range of voltages. The real world is analog. Digital systems restrict voltages to two ranges, representing values we call 0 and 1. For example, 0 V may indicate a 0 and 5 V may indicate a 1. Computers are digital systems. Digital systems are easier to design and more robust against noise, so much of the world has become digital. For example, analog cameras, televisions, and cell phones have been replaced by digital in the past decade or two. This improves reception, drives down cost, and increases capabilities.

The smallest piece of information in a computer is a single 0 or 1. This is called a binary digit, or *bit* for short. A group of 8 bits is called a byte (B). 2^{10} = 1024 bytes is called a kilobyte (KB).

A group of N bits can represent 2^N possibilities. We often use them to represent a number in the range [0, 2^N-1]. For example, an 8-bit number could represent a value from 0-255. A 16-bit number represents a value from 0-65535. A 32-bit number represents a value from 0 to 4 billion. Later, we will discuss alternative ways to use an N-bit value to represent negative or noninteger numbers.

When digital systems interact with the real world, the analog voltages must be converted to digital values. This is done by an analog-to-digital (A/D) converter, which accepts an analog voltage and produces a digital value proportional to the input voltage.

2. ARDUINO

The Mudduino is a custom version of the open-source Arduino microcontroller platform. Arduino makes it easy and cheap to build systems involving both hardware and software so that programs can interact with the real world. The Arduino project was started by a team of Italians in 2005 but has spread worldwide. It is extremely popular with universities and hobbyists.

An Arduino system includes a microcontroller, multiple analog and digital inputs and outputs, a power supply, and a way to interface with a host computer.

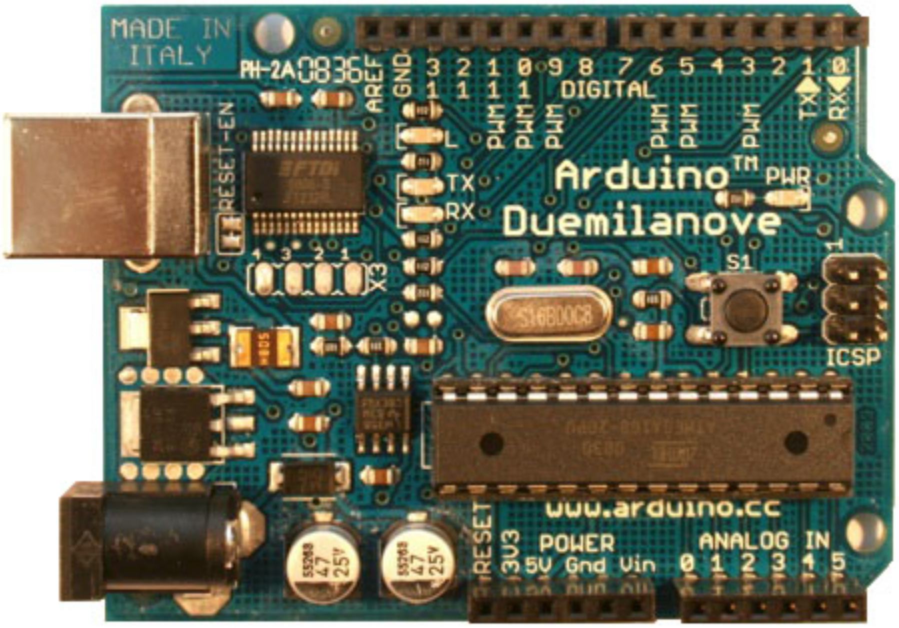


Figure 1 Arduino Duemilanove Board

Source: www.arduino.cc

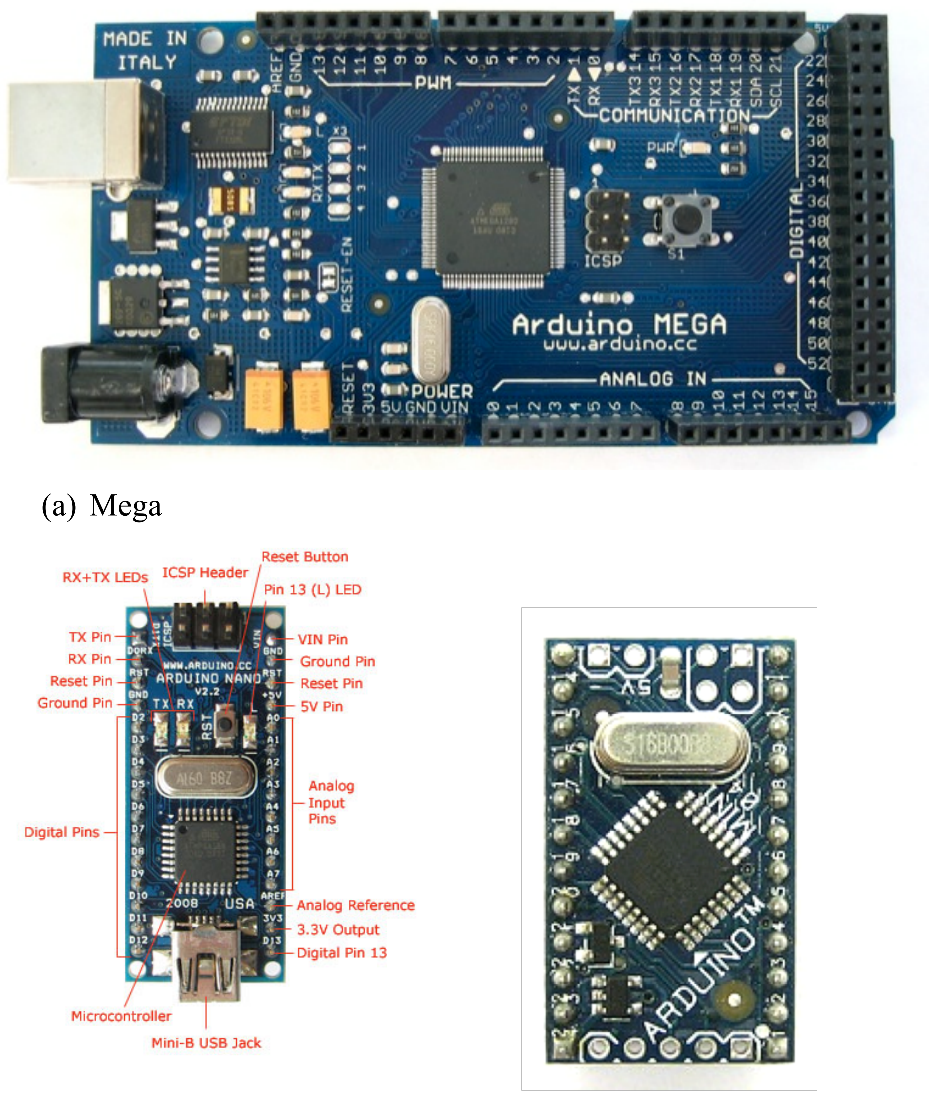
2.1. OFFICIAL ARDUINO BOARDS

You can buy a variety of official Arduino boards. The designs are open-source, so you could also download the printed circuit board files, get your own manufactured, buy the components, and assemble it yourself. It's typically cheaper to buy one off the shelf unless you are making hundreds.

The most common Arduino board in 2010 is the Duemilanove. Duemilanove means 2009 (the year it was introduced) in Italian. Table 1 summarizes capabilities of other official boards and Figure 2 shows photographs of the boards.

Table 1 Official Arduino Boards

Version	Digital I/Os	Analog Inputs	Price	Notes
Duemilanove	14	6	$28	Very popular
Mega	54	16	$65	More memory and I/Os
Nano	14	8	$35	0.73” x 1.70”, fits on a breadboard
Mini	14	8	$25	Smallest (0.7” x 1.3”), fits on breadboard, requires Mini USB adapter for programming



(a) Mega

(b) Nano

(c) Mini

Figure 2 Arduino boards

Source: www.arduino.cc

2.2. POWER SUPPLY

The Arduino system operates at 5 V.

When plugged into a USB cable, the system will draw a steady 5 V directly from USB. A fuse on the board will blow if the system tries to draw more than 500 mA from the USB port, preventing the Arduino from damaging the host computer if a short circuit occurs.

Alternatively, the Arduino can draw power from a DC jack so that it can operate off a battery pack or wall transformer. The input voltage should be in the range of 7 – 12 V. An on-board regulator drops this to a steady 5 V.

2.3. INPUTS AND OUTPUTS

One of the attractions of Arduino is easy access to digital and analog input/output (I/O) pins.

An Arduino board has at least 14 digital pins that can be configured as inputs or outputs. The digital signals use the following logic levels:

Table 2 Arduino Input and Output Voltage Levels

Level	**Value**	**Notes**
V_{IL}	1.5	Maximum input voltage recognized as a 0
V_{IH}	3.5	Minimum input voltage recognized as a 1
V_{OL}	0.9	Maximum output voltage produced for a 0
V_{OH}	4.2	Minimum output voltage produced for a 1

In other words, the Arduino digital pins will interpret voltages in the range of -0.5 – 1.5 V as a logic 0 and voltages in the range of 3.5 – 5.5 V as a logic 1. It will produce voltages in the range of 0 – 0.9 V for a 0 and 4.2 - 5 V for a 1. The broader input voltage range means that the system has some noise margin. If a valid output is distorted by a modest amount of noise, it still can be recognized as a valid input.

The output pins can drive a maximum of 40 mA of current. This is enough to operate a light-emitting diode, but not enough to directly drive a motor.

A portion of the output pins can be used in pulse-width modulation mode. Pulse width modulation gives the ability to simulate an analog output using a digital output pin. In this mode, the pin alternates between 0 and 5 V at about 490 Hz. Depending on the fraction of the time (e.g. the pulse width) at each voltage, the average output value takes on some value between 0 and 5 V. For example, the pulse-width modulation output can be connected to flash an LED on and off. Because the eye is not sensitive to changes faster than about 70 Hz, the eye will perceive an average brightness somewhere between fully ON and fully OFF.

The Arduino also has a limited number of pins that can be used as analog inputs. These pins are connected to a 10-bit A/D converter. The A/D converter produces a number proportional to the analog voltage, with 0 V corresponding to 0 and 5 V corresponding to the maximum value of 1023. A/D conversion takes about 100 microseconds, so the maximum sampling rate is about 10,000 samples / second.

2.4. HOST INTERFACE

The Arduino communicates with a host computer over a USB cable. The user develops software on the host computer and uploads it to the Arduino. The Arduino program can print messages or send data back over the USB cable to a terminal running on the host computer.

The Arduino processor internally uses an RS232 serial interface. A converter chip converts data back and forth between USB and RS232 standards.

RS232 serial communication involves sending a packet consisting of a start bit, 8 data bits, and a stop bit. Thus, 10 bits are communicated for every data byte received. The baud rate is the number of communicated bits per second. Thus 9600 baud means 9600 bits/second, or 960 data bytes per second.

The host and Arduino must agree on the rate at which data is being sent. For historical reasons, the default rate is 9600 baud. You can change this to rates as high as 115200 baud if you need to communicate faster, but must adjust it in both your program and the terminal.

3. MICROCONTROLLER

The Arduino systems use microcontrollers from Atmel. A microcontroller is a microprocessor with additional features to more easily interact with the real world. The microprocessor is the brain of a computer. It executes a sequence of instructions called a program. The Atmel processors are 8-bit microcontrollers, meaning that they operate on data in chunks of 8 bits at a time. Operating on a 32-bit number would require a sequence of four operations. 8-bit microcontrollers are inexpensive. For example, our chip costs about $2.32 in quantities of 2000 or more. Some competing microcontrollers cost as little as $0.50.

We will be using the Atmega328 microcontroller. Its features are described below. Most competing microcontrollers have similar features.

- 16 MHz clock, executes one instruction per clock cycle
- Analog and digital input/output pins (including a built-in A/D converter)
- On-chip memory for holding programs and data
    - 32 KB of Flash memory for programs (retains value after power-off)
    - 2 KB of static random access memory (SRAM) for data (looses value when powered down)
- Multiple timers/counters for measuring elapsed time and generating periodic signals
- Universal Asynchronous Receiver and Transmitter (UART) serial port controller
- Interrupts to change the flow of program operation when an event occurs

For more information, look up the ATmega328 datasheet on the web:

http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf

4. MUDDUINO BOARD

The Mudduino board shown in Figure 3 is a customized version of the Arduino developed by Matthew Keeter and Madeleine Ong at HMC for E11. It has similar capabilities to the Duemilanove. However, it mostly uses throught-hole components that are easy to solder so that you can assemble your own board. It also offers connectors customized to robotics applications, a form factor designed for your vehicle, and a blank portion where you can add your own circuitry.

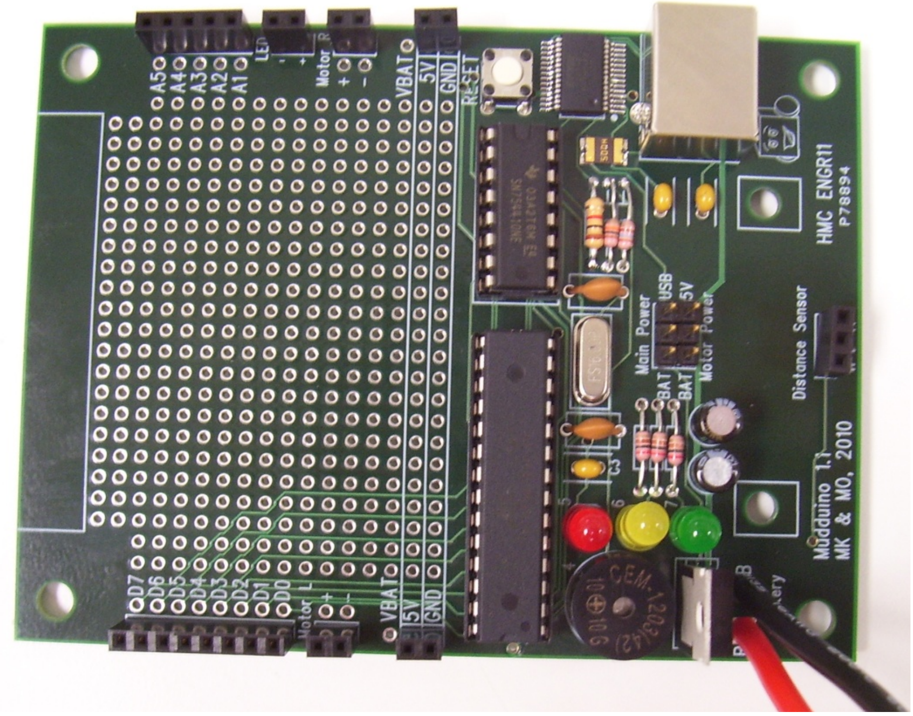


Figure 3 Assembled Mudduino Board

4.1. FEATURES

Major features of the Mudduino Board include:

- Atmega328 microcontroller with 32 KB of Flash program memory and 2 KB of RAM data memory operating at 16 MHz
- H Bridge for driving two high-current motors
- Three indicator LEDs (red, yellow, green)
- Buzzer
- USB connector to communicate with a host
- Power supply circuitry
- Battery connector for unteathered operation
- Expansion area for soldering on custom hardware

- Header pins for connecting:
    - 8 digital ports
    - 5 analog ports
    - 2 motor ports
    - Distance sensor port
    - A power indicator / identifier LED
    - 5 V and GND
    - 20 expansion pins

4.2. SCHEMATIC

Figure 4 shows a schematic of the Mudduino board.

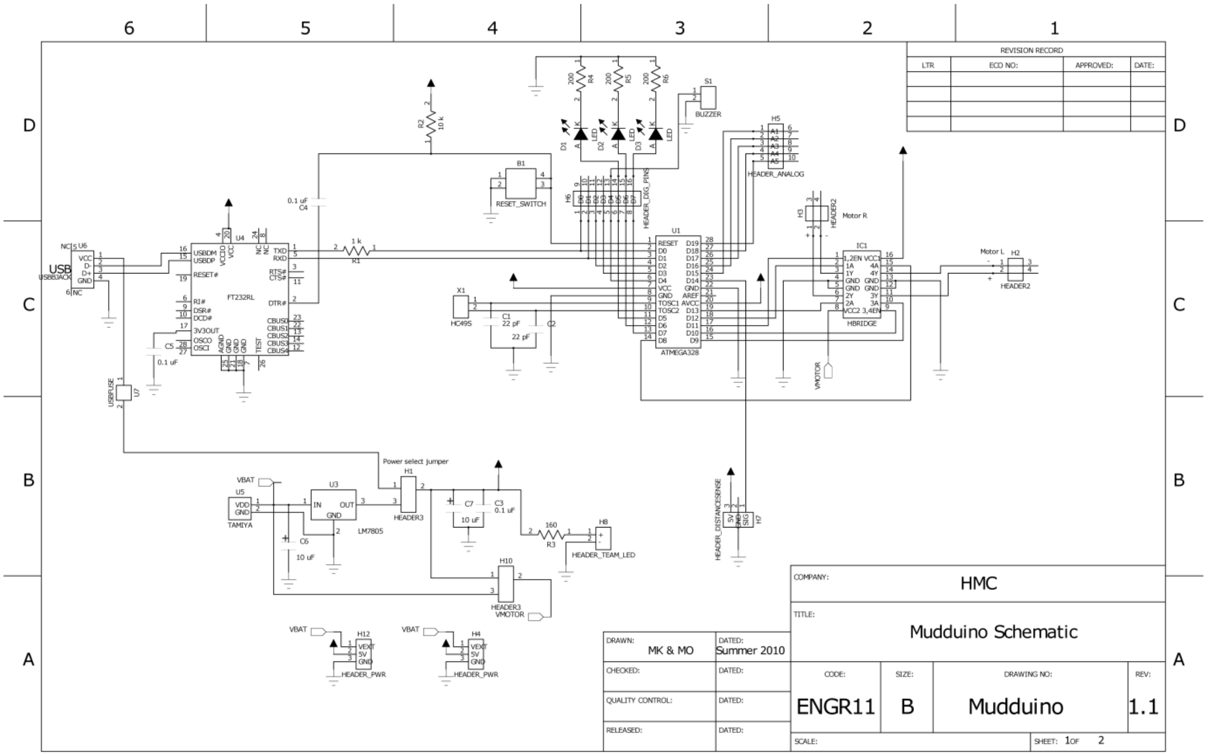


Figure 4 Mudduino Schematic

The ATmega328 microcontroller is at the heart of the board. The pins primarily connect to I/O ports as will be summarized in the next section.

Two pins connect to a 16-MHz crystal and a pair of capacitors. The microcontroller resonates this crystal to make a 16 MHz clock.

The reset button pulls the reset pin low, causing the microcontroller to restart operation at the beginning of its program.

The power supply circuitry includes jumpers to select whether power should come from the battery or USB port. The battery voltage goes through a 5 V regulator to drop down to the desired level. The USB port provides 5 V directly; a fuse is included to protect the host computer from current overload. Capacitors on the power supply help filter out noise, providing a stable power source. The voltage for the motors can also come from either the battery or 5 V supply. Don't run the motors off the USB port; if they stall, they can draw so much current that they will blow the fuse.

The microcontroller includes a conventional serial port, yet the host interface uses a USB cable. The FT232RL USB to serial bridge performs the conversion. The host can upload programs over the serial link to the microcontroller, and the microcontroller can communicate with a terminal on the host over the port. The bridge also generates a reset signal to restart the microcontroller after programming is complete.

4.3. I/O PINS

Table 3 summarizes the I/O pins used on the board.

Table 3 Mudduino I/O pins

Digital Pin #	Analog Pin #	Notes
0		Header D0, Serial TX
1		Header D1, Serial RX
2		Header D2
3		Header D3
4		Header D4, Buzzer
5		Header D5, Red LED
6		Header D6, Yellow LED
7		Header D7, Green LED
8		Left Motor Black
9		Left Motor Red
10		Left Motor Enable
11		Right Motor Enable
12		Right Motor Black
13		Right Motor Red
14	0	Distance Sensor
15	1	Header A1
16	2	Header A2
17	3	Header A3
18	4	Header A4, Phototransistor
19	5	Header A5, Reflectance Sensor

Note that all 20 pins can be used for digital inputs and outputs. Six of the pins can also be used as analog inputs.

Digital pins 3, 5, 6, 9, 10, and 11 can be used to simulate analog outputs by pulse-width modulating the digital pins using built-in circuitry.

Pins 2 and 3 support external interrupts that notify the program when the value on these pins change.

4.4. BOOTLOADER

Your ATmega328 has been preprogrammed with the Arduino bootloader. This program occupies part of the Flash memory on the chip. It initializes the chip at reset or powerup and monitors the serial port, waiting for a program to be uploaded.

The bootloader toggles some of the I/O pins during startup. This has the unfortunate side effect of making the right motor briefly spin.

5. BOARD ASSEMBLY

You will solder together your own Mudduino board in lab. Solder is a 60/40 alloy of tin and lead (Sn/Pb) with a melting point of 370 °C. It is heated with a soldering iron to melt and flow to form a joint with good electrical and mechanical properties.

5.1. PRINTED CIRCUIT BOARDS

You will start with a printed circuit board (PCB). The board contains four layers of copper sandwiched between a green insulating fire-retardant resin called FR4. The inner two layers of copper are nearly solid and are dedicated to carrying power (5 V) and ground. This provides a stable power supply and reduces noise problems (an earlier incarnation of the board used two layers and would crash when the motors started and drew spikes of current that disturbed the power supply). The outer two layers are used for routing signals across the board. Tiny copper-plated holes called vias are used to connect between layers of the board.

5.2. THROUGH-HOLE VS. SURFACE MOUNT

Electrical components are classified as through-hole or surface mount. Through-hole components stick through a copper-plated hole in the board and are soldered on the reverse side. Components with a row of pins normally place the pins 0.1” apart, making them fairly easy for a novice to solder, but taking up a large amount of room. Surface mount components are soldered to the surface of the board. Because no holes are required, the pins may be spaced more closely together.

Surface mount (SMT) components are usually soldered by applying some solder paste to the pads on the board, placing the component on the pads and aligning it carefully, then applying hot air to melt the paste. Surface tension causes the solder to adhere to the pins and usually prevents bridges from forming between pins. While surface-mount soldering is not difficult with the right equipment and training, your board will come with a few SMT components already attached so you won’t have to learn on this project.

5.3. *BOARD LAYOUT*

The top of the PCB is called the component side. The components are mounted on this side. The reverse is called the solder side. The through-hole components are soldered on the solder side. Figure 5 shows a plot of the component side of the Mudduino PCB.

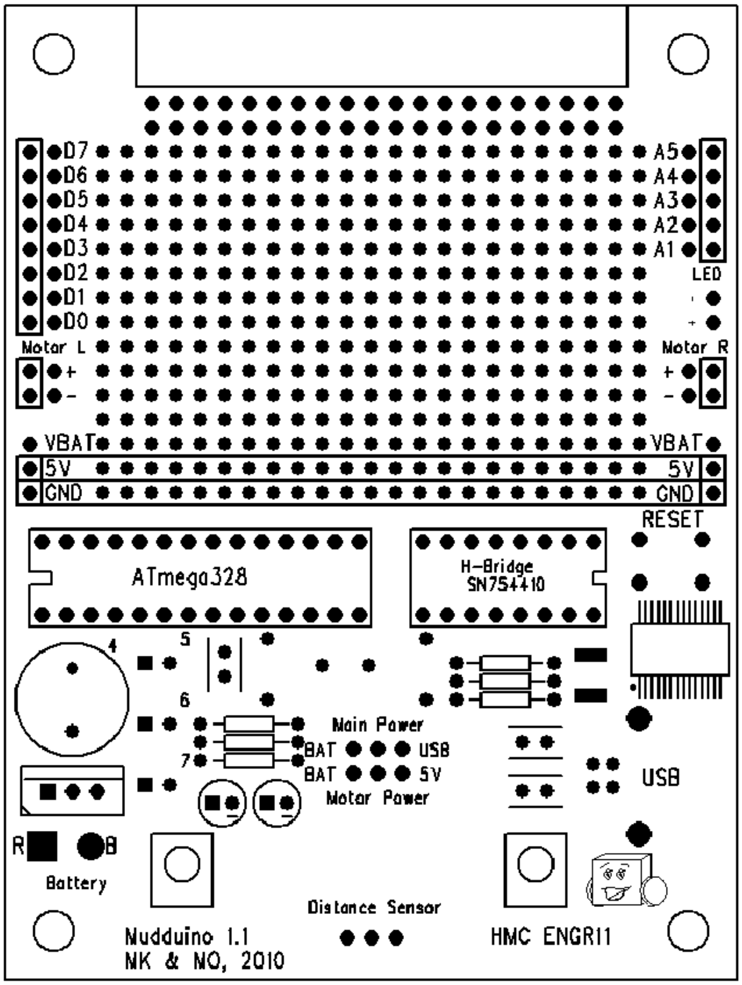


Figure 5 Mudduino Printed Circuit Board

5.4. *COMPONENTS*

You will need to learn to identify components to place them on the board. Most components are fairly simple, but resistors and capacitors have codes you'll need to learn to read.

5.4.1. Resistors

Resistors are identified with four color bands. The first three indicate the value, in ohms (Ω). The fourth indicates the tolerance; it is typically gold, indicating +/- 5%. The tolerance can usually be ignored unless you are concerned about precision circuits.

The values are determined using the following color codes:

Table 4 Resistor Color Codes

0	Black
1	Brown
2	Red
3	Orange
4	Yellow
5	Green
6	Blue
7	Violet
8	Grey
9	White

This can be remembered using the mnemonic "**B**ad **b**eer **r**ots **o**ur **y**oung **g**uts **b**ut **v**odka **g**oes **we**ll" or by other even less politically correct sayings.

The first two bands are read as a two digit number, and the third is a power of ten multiplying the number. For example, red – black – brown – gold indicates 201, meaning 20 x 10^1 = 200 Ω, with a 5% tolerance. Brown – black – orange – gold indicates 103, meaning 10 x 10^3 = 10,000 Ω (written 10 kΩ).

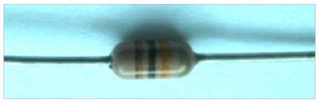
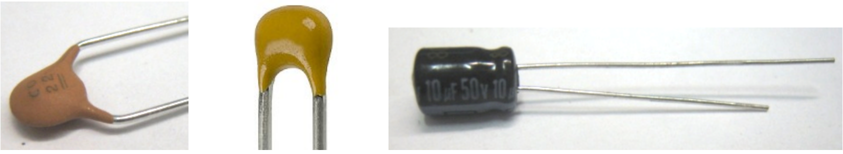


Figure 6 10 kΩ resistor

5.4.2. Capacitors

Capacitors are notoriously difficult to read. Usually large-valued electrolytic capacitors have their value written on the side. For example, a 10-microfarad (10^{-3} F) capacitor is labeled 10 μF. Small capacitors are sometimes labeled with a 3-digit code indicating their value in picofarads (10^{-12} F) in a fashion similar to resistors. For example, 104 means 10 x 10^4 pF = 10^{-7} F = 0.1 μF. A 2-digit code just indicates the value in picofarads. For example, 22 means 22 pF.



Ceramic – 22pF Ceramic – 0.1uF Electrolytic – 10uF

Figure 7 Capacitors