Lecture 3
Architecture of Arduino development board

http://www.arduino.cc/
What do you need to start working with Arduino?

1. Arduino board
2. USB cable
3. Computer with USB interface
4. USB driver and Arduino application
to be downloaded from
(http://arduino.cc/en/Main/Software)
What is Arduino?

- The Arduino is a programmable hardware board that runs an 8-bit /16 Mhz microcontroller with a special bootloader that allows users to upload programs to the microcontroller.
  - It has digital input pins for input from switches and output to Actuators (LEDs or electrical motors)
  - It also has analog pins to accept inputs from voltage-based sensors.
- Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.
What is Arduino?/ Open Source

- **Open Source Hardware**
  
The Arduino system is open source - all hardware (made by Arduino distributors) has the schematics and PCB layouts available online.

- **Open Source Bootloader**
  
The bootloader is what runs on the chip before the program is run. It boots the chip and executes the program.

- **Open Source Development Kit**
  
The development kit - what you use to program an Arduino board - is also available online. It is free, open-source.
Arduino Terminology

- **I/O Board** - The I/O Board is the "brain" of the operation (main microcontroller you program it from your computer).

- **Shield** - A Shield is a device that plugs into an I/O Board. These extend the capabilities of the I/O Board.

- **Sketch** - A Sketch is a program written for the board and shields.
Arduino Terminology

**Sensor** - components (Gas, etc.)

**Modules** - serial data (GPS module, etc.)

**pin** – an input or output connected to something.

**Digital** – value is either HIGH or LOW.

**Analog** – value ranges, usually from 0-255.
Arduino Types

- Many different versions
  - Number of input/output channels
  - Processor
- Leonardo
- Due
- Micro/Nano
- LilyPad
- Esplora
- Uno/number one
Leonardo

- Compared to the Uno, a slight upgrade.
- Built in USB compatibility
- Presents to PC as a mouse or keyboard
Due

- Much faster processor, many more pins
- Operates on 3.3 volts
- Similar to the Mega
Micro/ Nano

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila has (electrically) with more analog input pins and onboard +5V AREF jumper.
LilyPad

The LilyPad Arduino is a microcontroller board designed for wearables and e-textiles. It can be sewn to fabric and similarly mounted power supplies, sensors and actuators with conductive thread.
Esplora

- Game controller
- Includes joystick, four buttons, linear potentiometer (slider), microphone, light sensor, temperature sensor, three-axis accelerometer.
- Not the standard set of IO pins.
Uno (means one)

- The pins are in three groups:
  - 6 analog inputs.
  - 14 digital input/output pins (of which 6 can be used as PWM outputs).
  - Power.
Shields

Shields connect to the I/O board to extend its functionality.

Wireless Network Shield

Color LCD Shield

GPS Shield

Power Driver Shield
More Shields…

Communication shields - XBee, Ethernet, and Wifi
Modules

Modules send serial data strings to the Arduino.

GPS Module

Bluetooth Module

Temperature & Humidity Sensor

RFID Module
Sensors and Modules

Shields aren't the only way to extend an Arduino board - you can hook sensors to it!

- Temp & Humidity
- Gas Sensor
- Fingerprint Scanner
- Flex Sensor
Sensors

Photoresistor, infrared, force sensitive resistor, Hall effect (varies its output voltage in response to a magnetic field), Piezo, tilt sensor.
# Arduino Uno/ATmega328

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microcontroller</strong></td>
<td>ATmega328</td>
</tr>
<tr>
<td><strong>Operating Voltage</strong></td>
<td>5 V</td>
</tr>
<tr>
<td><strong>Input Voltage</strong></td>
<td>7-12 V</td>
</tr>
<tr>
<td><strong>Input Voltage (limits)</strong></td>
<td>6-20 V</td>
</tr>
<tr>
<td><strong>Digital I/O Pins</strong></td>
<td>14 (of which 6 provide PWM output)</td>
</tr>
<tr>
<td><strong>Analog Input Pins</strong></td>
<td>6</td>
</tr>
<tr>
<td><strong>DC Current per I/O Pin</strong></td>
<td>40 mA</td>
</tr>
<tr>
<td><strong>DC Current for 3.3V Pin</strong></td>
<td>50 mA</td>
</tr>
<tr>
<td><strong>Flash Memory</strong></td>
<td>32 KB (ATmega328) of which 2 KB used by bootloader</td>
</tr>
<tr>
<td><strong>SRAM</strong></td>
<td>2 KB (ATmega328)</td>
</tr>
<tr>
<td><strong>EEPROM</strong></td>
<td>1 KB (ATmega328)</td>
</tr>
<tr>
<td><strong>Clock Speed</strong></td>
<td>16 MHz</td>
</tr>
</tbody>
</table>
Arduino Mega Board Overview
# Arduino Mega/ATmega2560

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATmega2560</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>54 (of which 14 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>16 (4 UARTs (hardware serial ports))</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>40 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>256 KB of which 8 KB used by bootloader</td>
</tr>
<tr>
<td>SRAM</td>
<td>8 KB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>4 KB</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
</tbody>
</table>
Nano
Nano

[Diagram of Arduino Nano with pin descriptions]
Components of the Arduino

• ATMega168/328.
• 16MHz crystal/filtering capacitors.
• Onboard power regulators.
• FTDI USB <-> Serial Chip.
• Hardware (Board).
Necessary parts for any circuit

• ATMega168/328.
• The ‘brains’ of the Arduino.
• Program is loaded onto the chip.
• Runs main loop until power is removed.
Timing

- 16Mhz Crystal.
- The ‘heartbeat’ of the ATMega chip.
- Speed of crystal determines chip speed.
- ATMega series has onboard oscillator.
Power Supply

- 5 Volt and 3.3 volt regulators.
- Filtering capacitors.
- Automatic switching between external and USB Power.
- Leave it out if you have a filtered 5 Volt power supply.
FTDI USB Chip

- Allows your Arduino to communicate with your computer over a simple USB link.
- ATmega16u2 replaces FT232RL for USB-serial communication / Arduino Uno R3
Hardware

- Circuit Board
- Headers
- USB port
- Sockets
Digital or Analog?

• Digital – may take two values only: ON or OFF (1 or 0)
• Analog – has many (infinite) values
IO Pins

Two states (binary signal) vs. multiple states (continuous signal)
Arduino Digital I/O

- `pinMode(pin_no., dir)`
  - Sets pin to either INPUT or OUTPUT

- `digitalRead(pin)`
  - Reads HIGH or LOW from a pin

- `digitalWrite(pin, value)`
  - Writes HIGH or LOW to a pin
Arduino Analog I/O

- `analogWrite(pin, value)`
  - `pin`: the pin to write to.
  - `value`: PWM the duty cycle: between 0 (always off) and 255.

- `int x = analogRead(A0);`
Pin Used as an Output

- Turn on LED, which is connected to pin Arduino pin 0 (PD0) (note the resistor!)

- What should the data direction be for pin 0 (PD0)?
  - `pinMode(0,OUTPUT);`
  - Turn on the LED
  - `digitalWrite(0,HIGH);`
  - Turn off the LED
  - `digitalWrite(0,LOW);`
Pins as Inputs and Pull-up Resistors - 1

- Using a switch as a sensor
- Ex. Seat belt sensor
- Detect the switch state
- What should the data direction be for Arduino pin 3 (PD3)?
  - pinMode(3, INPUT);
  - What will the voltage be on PD3 when the switch is closed?, 0v.
  - What will the voltage be on PD3 when the switch is open??
Switch as a sensor, cont.

Make the voltage on the pin determinate by turning on the pull-up resistor for PD3

Assuming PD3 is an input:

Turns on the “pull-up” resistor

- `pinMode(3,INPUT_PULLUP);`
- What will the voltage on PD3 be when the switch is open?, =5v
- What will the voltage on PD3 be when the switch is closed?, =0v
Switch as a sensor, cont.

To turn off the pull-up resistor, assuming PD3 is an input:

- `pinMode(3, INPUT)`
Pins as Inputs and Pull-up Resistors - 4

- Possibility of 'weak drive'
Pin Voltages

- Output pins can provide 40 mA of current
- HIGH or LOW (logic: 1 or 0)
- Voltages
  - **TTL**
    - 5 V (for HIGH)
    - 0 V (for LOW)
  - **3.3 V CMOS**
    - 3.3 V (for HIGH)
    - 0 V (for LOW)
The power pins are as follows:

- **Vin.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. (7v-12v)

- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from Vin via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.

- **GND.** Ground pins.
Serial Communication

- **RX** blinks when the Arduino is receiving data.
- **TX** blinks when the Arduino is transmitting data.
Serial data transfer with Arduino UNO

1. Universal asynchronous receiver/transmitter (UART)
2. Serial Peripheral Interface (SPI)
3. Inter-integrated Circuit (I2C) or Two-Wire Interface (TWI)
All Arduino boards have at least one serial port (also known as a UART). It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB.

Used for communication between the Arduino board and a computer or other devices.
Serial Peripheral Interface (SPI): is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices.

- One master device (usually a microcontroller) which controls the peripheral devices.

- Pins: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).

- Typically three lines common to all the devices:
  1. MISO (Master In Slave Out) - The Slave line for sending data to the master.
  2. MOSI (Master Out Slave In) - The Master line for sending data to the peripherals.
  3. SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master

- One line specific for every device:
  1. SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices. When a device's Slave Select pin is low, it communicates with the master. When it's high, it ignores the master.
I2C (I squared C)

- The Inter-integrated Circuit (I2C): is a protocol intended to allow multiple “slave” digital integrated circuits (“chips”) to communicate with one or more “master” chips.

- Pins: A4 or SDA pin and A5 or SCL pin
Arduino MEGA

- **Serial0**: 0 (RX) and 1 (TX); **Serial 1**: 19 (RX) and 18 (TX); **Serial 2**: 17 (RX) and 16 (TX); **Serial 3**: 15 (RX) and 14 (TX)

- **SPI**: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)

- **TWI**: 20 (SDA) and 21 (SCL)
SoftwareSerial

- It is possible to have multiple software serial ports (UART) with speeds up to 115200.

- If using multiple software serial ports, only one can receive data at a time.

- Not all pins on the Mega and Mega 2560 support change interrupts, so only the following can be used for RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69).