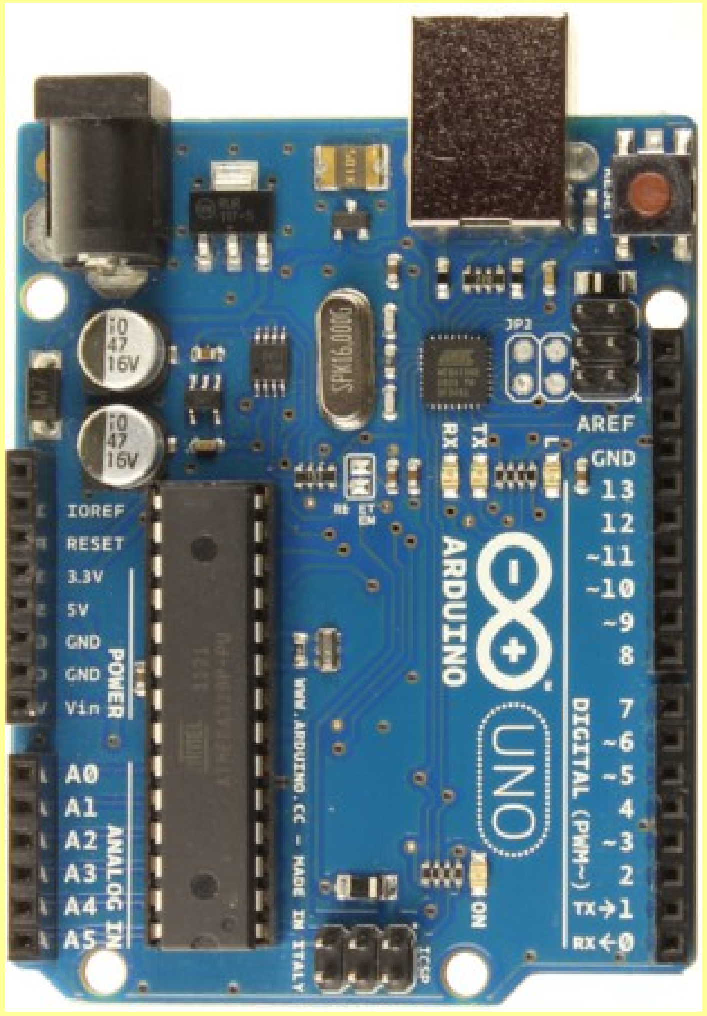
Using Arduino as a Hardware Programmer

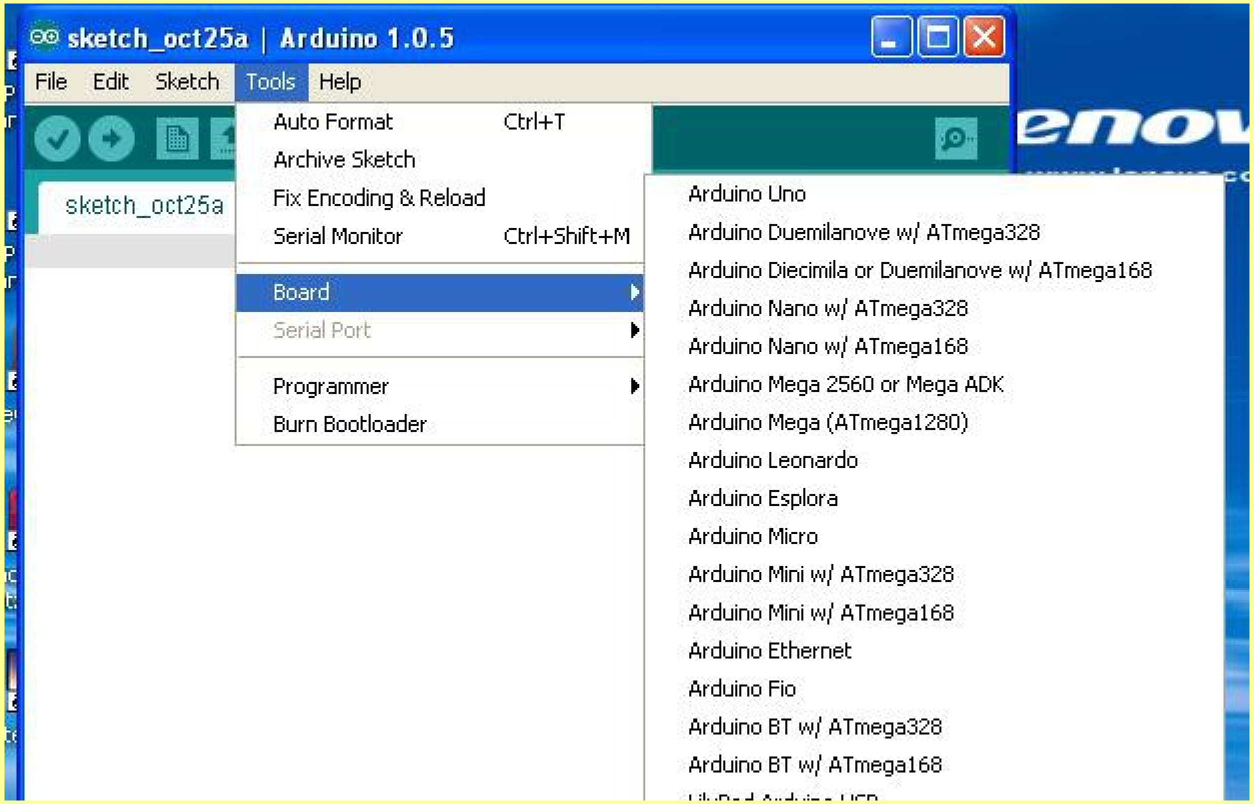
Dennis DeLorme
K-LUG Presentation
November 14 2013

Basic Arduino

- Hardware and software combination
- Oriented to ATmega328
- Extendability by use of configuration files
- Most complexity hidden from user
- Uses boot-loader capability of ATmega328
- Options to install boot-loader

Basic Arduino

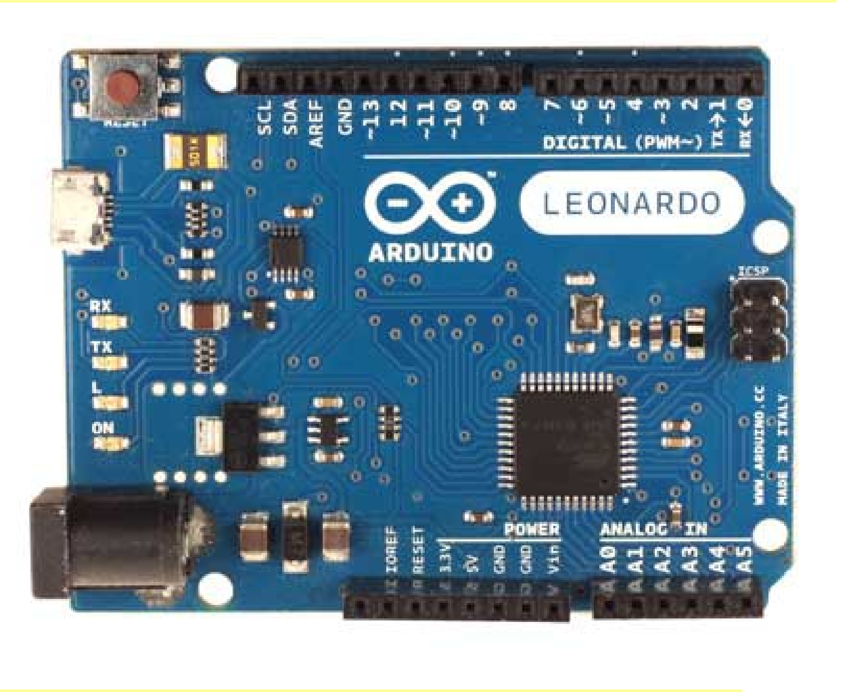


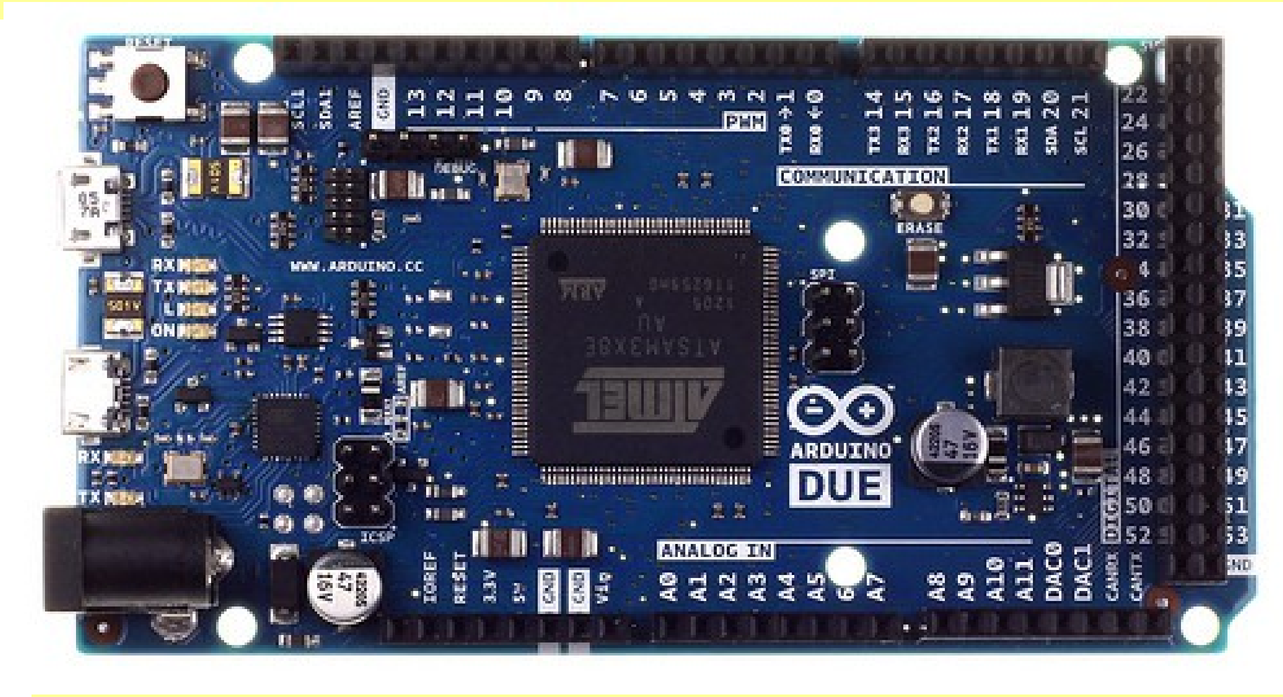


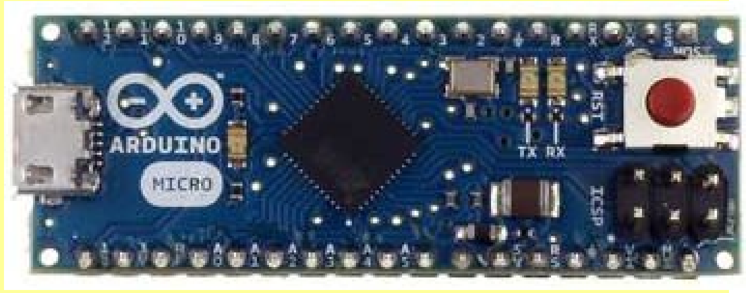
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader

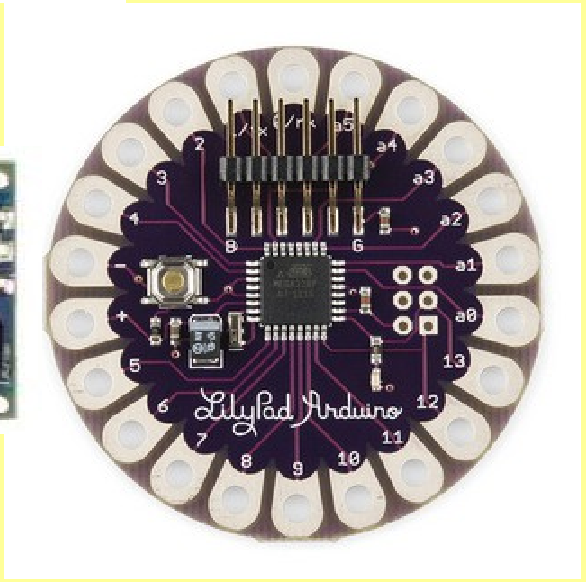
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168

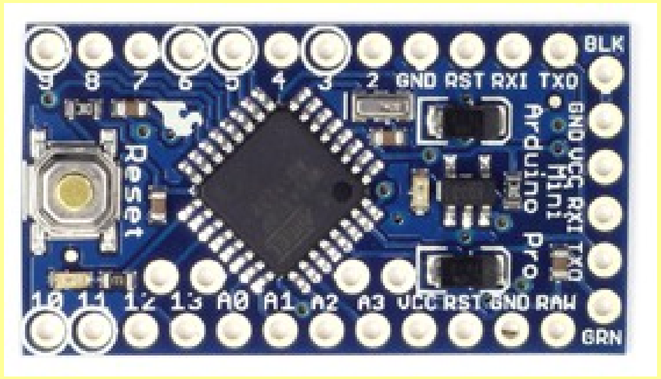
Some Mega Based Variants











Arduino Hardware Programming

- The Arduino contains a serial boot-loader in protected flash memory
- On reset the boot-loader is called
  - If serial handshake is successful new application code is downloaded and executed
  - If no serial handshake, existing application is executed

Arduino Hardware Programming

- The boot-loader is loaded into flash via the SPI interface using a special protocol
- An Arduino sketch can implement the SPI protocol so that an Arudino can program the boot-loader into an unprogrammed ATmega32 http://arduino.cc/en/Tutorial/ArduinoISP
- This same sketch can be used to program the flash on other AT chips – in our case the ATtiny 85

Atmel ATtiny Chips

- The ATtiny is a series of low cost, fewer features chips
- We will look at the 8 pin ATtiny85 family http://www.atmel.com/Images/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf
- These have
  - 2K/4K/8K of flash and 128/256/512 bytes of ram
  - 2 8 bit timers (PWM), ADC, USI (for SPI and I^2C)
  - No async serial support

Programming an ATtiny

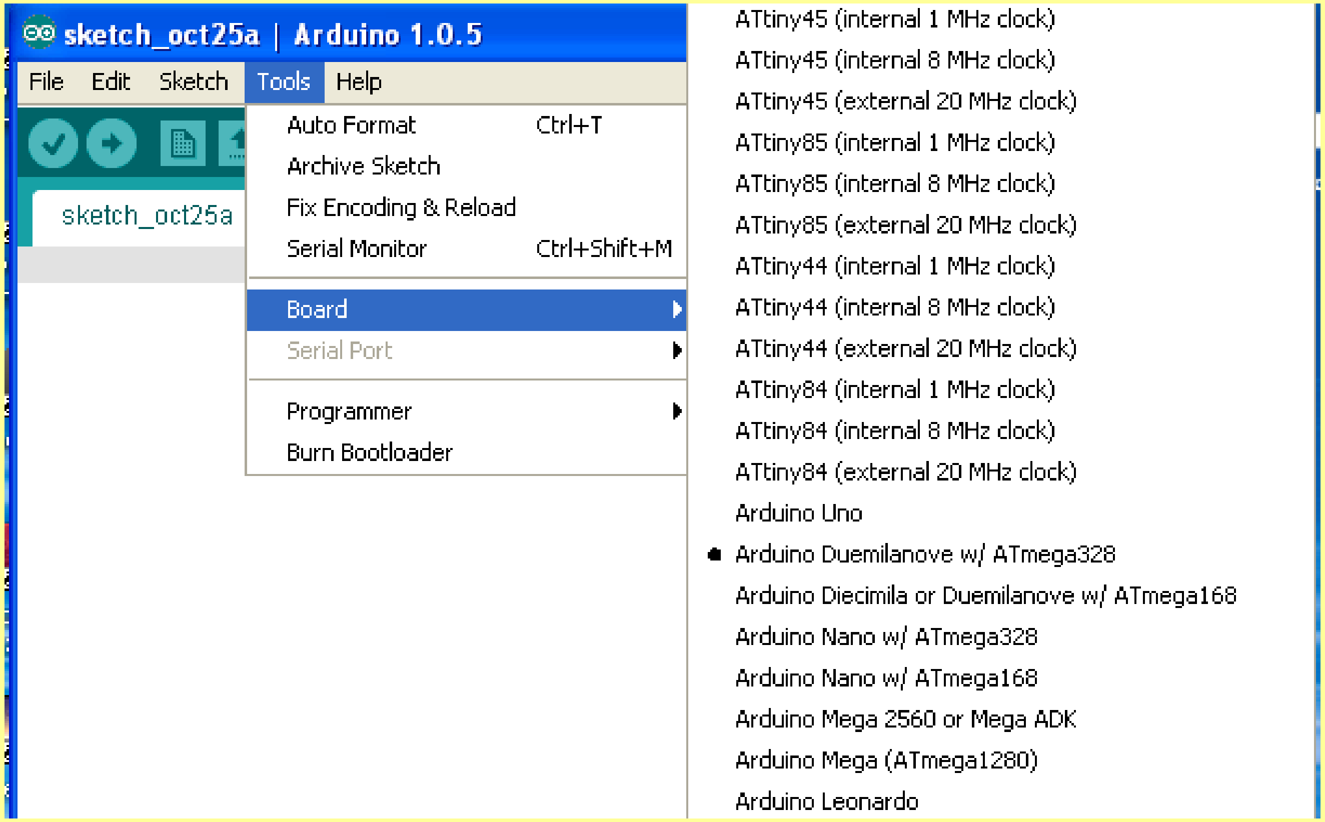
- Remember the Arduino environment is driven by text configuration files
- We need a file to to describe the chip features to the complier
- We need a file to define the chip hardware to the hardware programmer
- This work as been done – all we need to do is download and use it!

Programming an ATtiny

- Details are covered at http://hlt.media.mit.edu/?p=1695
- Download the ATtiny master.zip file and put the attiny directory in a directory named hardware in your sketch directory
- This will add the configuration files needed to develop code for the ATtiny84A and ATtiny85 chips.

Programming an ATtiny

- Start the Arduino IDE and you will see new options



sketch_oct25a | Arduino 1.0.5

File Edit Sketch Tools Help

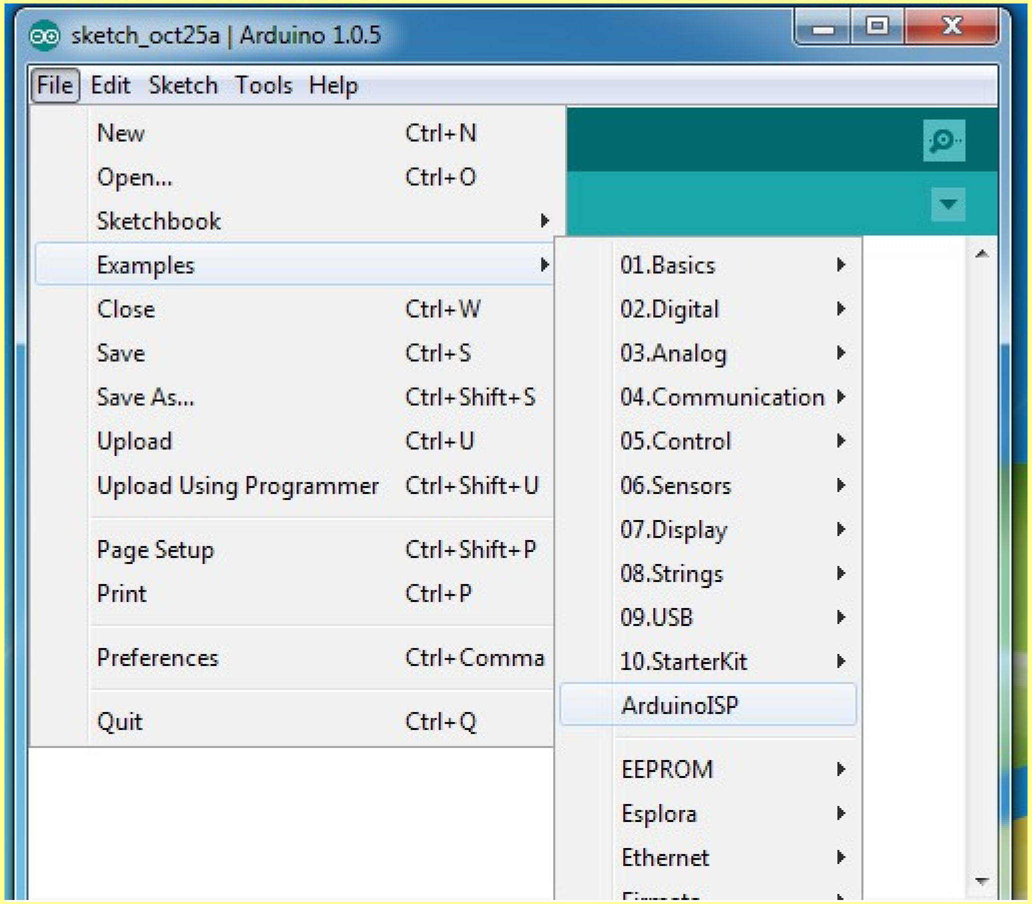
sketch_oct25a

Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader

ATtiny45 (internal 1 MHz clock)
ATtiny45 (internal 8 MHz clock)
ATtiny45 (external 20 MHz clock)
ATtiny85 (internal 1 MHz clock)
ATtiny85 (internal 8 MHz clock)
ATtiny85 (external 20 MHz clock)
ATtiny44 (internal 1 MHz clock)
ATtiny44 (internal 8 MHz clock)
ATtiny44 (external 20 MHz clock)
ATtiny84 (internal 1 MHz clock)
ATtiny84 (internal 8 MHz clock)
ATtiny84 (external 20 MHz clock)
Arduino Uno
● Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo

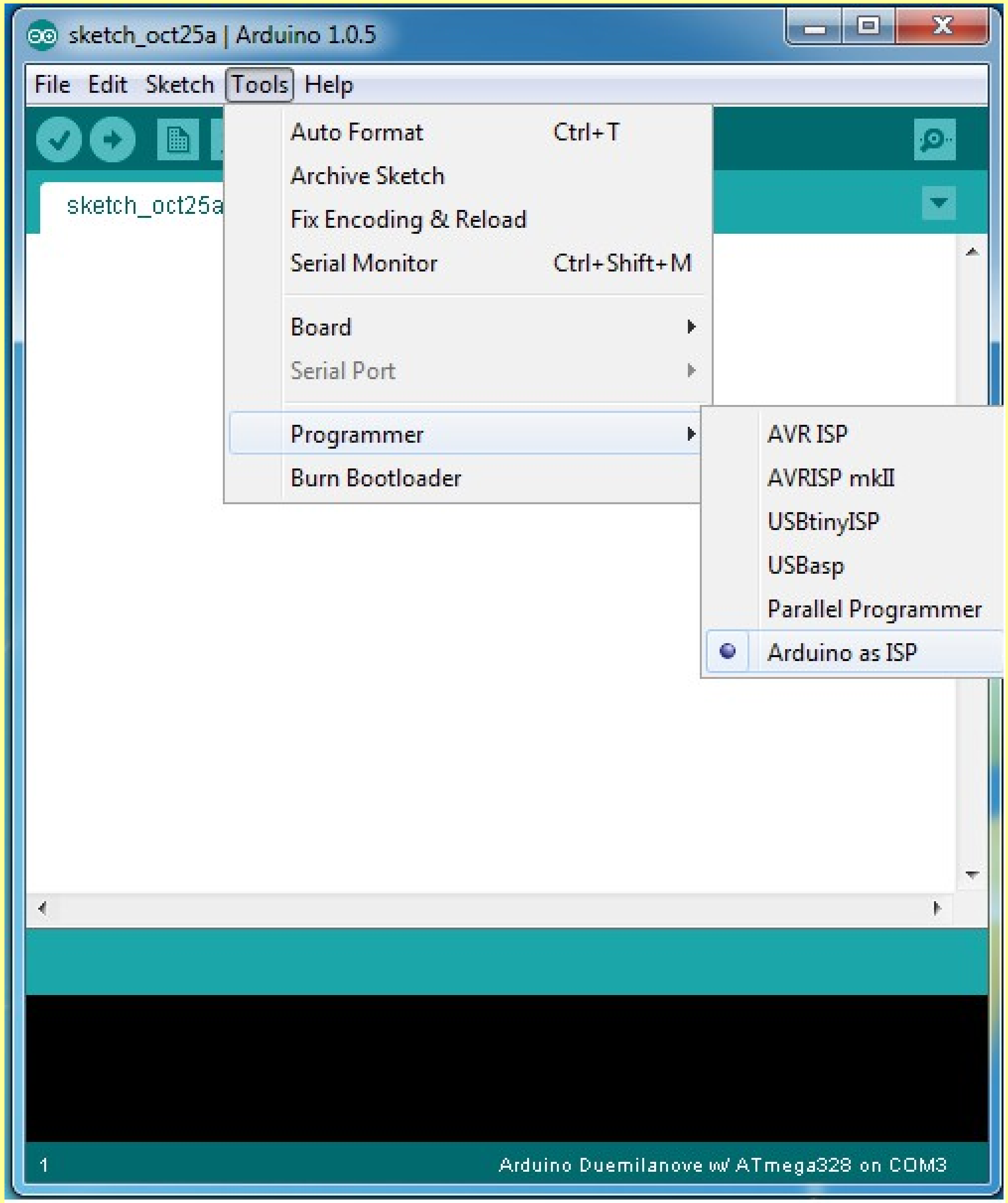
Programming an ATtiny

- You need to program an Arduino to be the flash loader
- Download this sketch into your Arduino



Programming an ATtiny

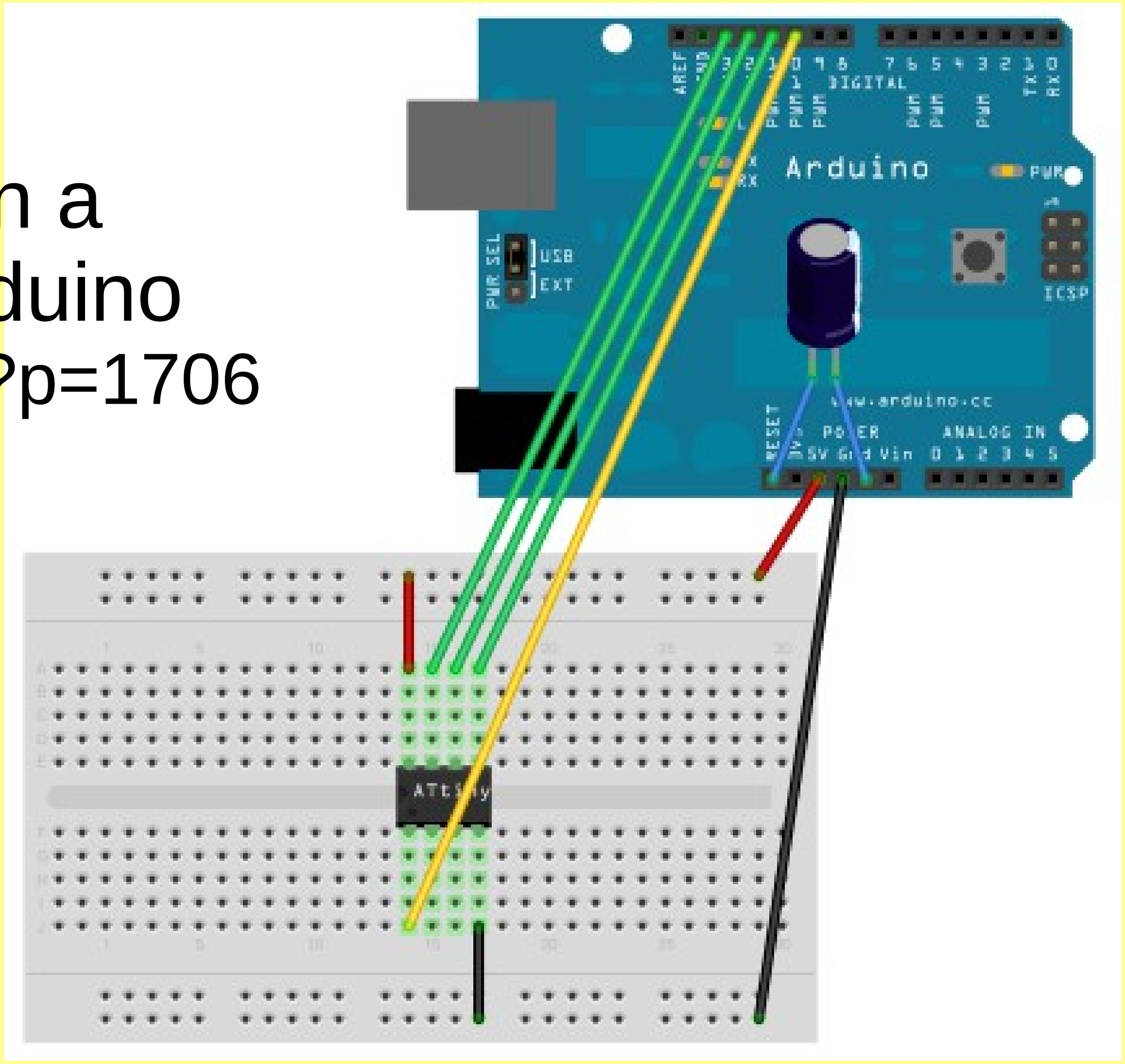
- Select the Arduino as the flash programmer



Programming an ATtiny

- Connect the ATtiny on a breadboard to the Arduino http://hlt.media.mit.edu/?p=1706

Note: You may need a ~10uf cap from reset to ground



Programming an ATtiny

- Pick the ATtiny chip you want to use from the Tools → Board options.
- Depending on the options you picked for the ATtiny you may have to set the hardware configuration bits
- To set the configurations bits click Tools → Burn Bootloader
- You only have to do this once
- For the ATtiny this doesn't burn a bootloader – it just sets the configuration bits

Programming an ATtiny

- Write your code for the ATtiny within the limits supported by the chip.
- Click the upload icon to load your code into the ATtiny
- Disconnect the programming jumpers and power
- Attach your peripherals and power up to test

Leonardo Considerations

- See http://petervanhoyweghen.wordpress.com/2012/09/16/arduinoisp-on-the-leonardo/
- The Leonardo does not have the SPI interface on the digital pins so the 6 pin ICSP header must be used
- The SS signal doesn't exist
  - In the ArduinoISP.ino sketch change

    ```
    #define RESET SS
    ```

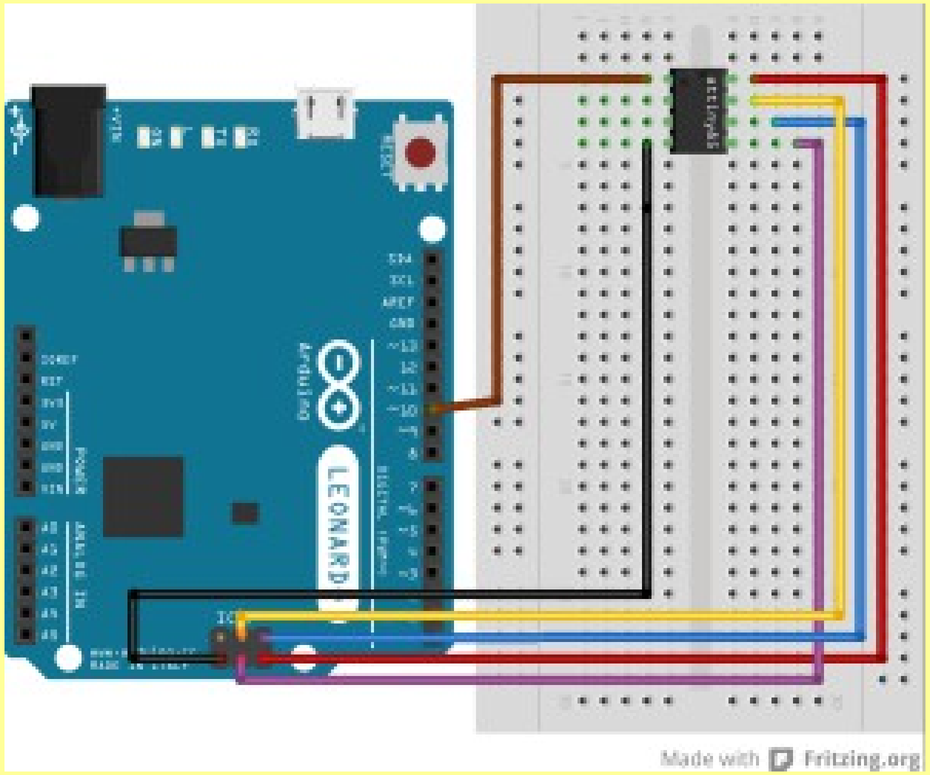
    into:

    ```
    #define RESET 10
    ```

Leonardo Considerations

The Leonardo connections to the ICSP header and digital pin 10 for reset

These will also work for original Arduino versions



Leonardo - Got Windows?

- You are not finished yet – the default programming protocol doesn't work – see previous link
- Add the following as programmers.txt to a leofix folder in your hardware folder

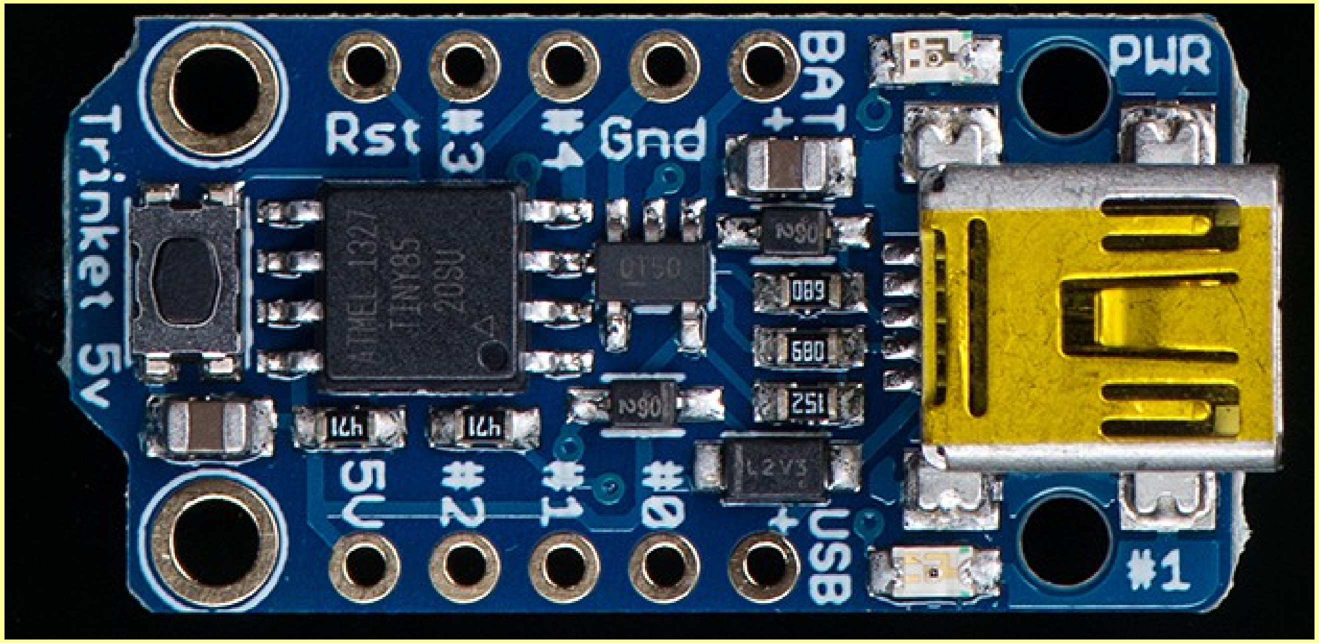
```
arduinoispleo.name=Arduino as ISP (Leonardo)
arduinoispleo.communication=serial
arduinoispleo.protocol=arduino
arduinoispleo.speed=19200
```

- When you restart the IDE there will be new entry in Tools → Programmer

  Arduino as ISP (Leonardo)

Is All This Too Much Work?

- Go to http://www.adafruit.com/category/167 and get a Trinket for about $8
- This is an ATtiny85 with a USB bootloader – but there are some limitations



IR Remote Demo

- The ATtiny85 will be used to decode the NEC protocol and modify I/O pins based on the IR remote signals
- The NEC protocol gives an 8 bit device code and an 8 bit command code for a button press on the remote
- An IR receiver chip is used to demodulate the IR signal and pass the raw digital data stream to the ATtiny85

IR Remote Demo

- The ATtiny85 will decode the serial digital data stream and extract the device and command codes
- Selected command codes will control I/O pins
- The SoftwareSerial library will be used to send the device and command codes to the PC via a USB to serial cable.