Arduino

Digital I/O
- pinMode(pin, mode)
- digitalWrite(pin, value)
- int digitalRead(pin)

Analog I/O
- analogReference(type) DEFAULT (5V), INTERNAL (1.1V?), EXTERNAL (AREF)
- int analogRead(pin) 10-bit
- analogWrite(pin, value) - PWM

Advanced I/O
- shiftOut(dataPin, clockPin, bitOrder, value)
- unsigned long pulseIn(pin, value)

Time
- unsigned long millis()
- unsigned long micros()
- delay(ms) ms,us: unsigned long
- delayMicroseconds(us) 3-16383us accurate, stops interrupts

Math
- min(x, y)
- max(x, y)
- abs(x)
- constrain(x, a, b)
- map(value, fromLow, fromHigh, toLow, toHigh)
- pow(base, exponent)
- sqrt(x)

Trigonometry
- sin(rad)
- cos(rad)
- tan(rad)

Random Numbers
- randomSeed(seed)
- long random(max)
- long random(min, max)

Bits and Bytes
- byte lowByte(x)
- byte highByte(x)
- 1/0 bitRead(x,n)
- bitWrite(x,n,b) n from LSB
- bitSet(x,n) sets to 1
- bitClear(x,n) sets to 0
- bit(n) returns 2^n

External Interrupts
- attachInterrupt(interrupt, function, mode)
- detachInterrupt(interrupt)

Interrupts
- interrupts()
- noInterrupts()

Serial

Inputbuffer is 128 bytes
- begin(baud)
- available()
- read()
- flush()
- print(data)
- println(data)
- write(data)
- write(str)
- write(buf,len)

EEPROM
- byte EEPROM.read(address)
- EEPROM.write(address, value)
  - address: int 0-511

Datatypes

char (signed)	8 bits
byte (unsigned)	8 bits
int	16 bits
word (unsigned)	16 bits
long	32 bits
float	32 bits
double	32 bits

Firmata

Methods
- begin() start the library
- begin(long) start the library and override the default baud rate
- printVersion() send the protocol version to the host computer
- blinkVersion() blink the protocol version on pin 13
- printFirmwareVersion() send the firmware name and version to the host computer
- setFirmwareVersion(byte major, byte minor) set the firmware name and version, using the sketch's filename, minus the .pde

Sending Messages
- sendAnalog(byte pin, int value) send an analog message
- sendDigitalPorts(byte pin, byte firstPort, byte secondPort) send digital ports as individual bytes
- sendDigitalPortPair(byte pin, int value) send digital ports as one int
- sendSysex(byte command, byte bytec, byte* bytev) send a command with an arbitrary array of bytes
- sendString(const char* string) send a string to the host computer
- sendString(byte command, const char* string) send a string to the host computer using a custom command type

Receiving Messages
- available() check to see if there are any incoming messages in the buffer
- processInput() process incoming messages from the buffer, sending the data to any registered callback functions
- attach(byte command, callbackFunction myFunction) attach a function to an incoming message type
- detach(byte command) detach a function from an incoming message type

Callback Functions

In order to attach your function to a message type, your function must match the standard callback function. There are currently three types of callback functions in Firmata: generic, string, and sysex.

- generic void callbackFunction(byte pin, int value);
- system_reset void systemResetCallbackFunction(void);
- string void stringCallbackFunction(char *myString);
- sysex void sysexCallbackFunction(byte pin, byte byteCount, byte *arrayPointer);

Message Types

These are the various message types that you can attach functions to.

- ANALOG_MESSAGE the analog value for a single pin
- DIGITAL_MESSAGE 8-bits of digital pin data (one port)
- REPORT_ANALOG enable/disable the reporting of analog pin
- REPORT_DIGITAL enable/disable the reporting of a digital port
- SET_PIN_MODE change the pin mode between INPUT/OUTPUT/PWM/etc.
- FIRMATA_STRING C-style strings, uses stringCallbackFunction for the function type
- SYSEX_START generic, arbitrary length messages (via MIDI SysEx protocol), uses sysexCallbackFunction for the function type
- SYSTEM_RESET message to reset firmware to its default state, uses systemResetCallbackFunction for the function type