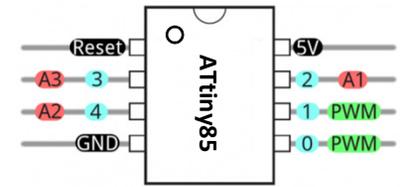




Sparkfun Electronics ATtiny85 Arduino Quick Reference Sheet



STRUCTURE

```
/* Each Arduino sketch must contain the following two functions. */
```

```
void setup()
{ // this code runs once at the beginning of the code execution.
}
```

```
void loop()
{ // this code runs repeatedly over and over as long as the board is powered.
}
```

COMMENTS

```
// this is a single line comment
/* this is a multiline comment */
```

SETUP

```
pinMode(pinNum, INPUT/OUTPUT/INPUT_PULLUP);
/* Sets the mode of the digital I/O pin. All pins are general I/O on the board. You must define what the pin will be used for at the beginning of your code in setup() */
```

CONTROL STRUCTURES

```
if(condition)
{ // if condition is true, do something here
}
else
{ // otherwise, do this
}
```

```
for(init; condition; increment)
{ // do this
}
```

DIGITAL I/O

```
digitalWrite(pin, val);
```

```
/* val = HIGH or LOW write a HIGH or a LOW value to a digital pin. */
```

```
buttonVal = digitalRead(pin);
```

```
/* Reads the value from a specified digital pin, either HIGH or LOW. */
```

ANALOG I/O

```
analogWrite(pin, val);
```

```
/* Writes an analog value to a pin. val = integer value from 0 to 255 */
```

```
sensorVal = analogRead(pin);
```

```
/* Reads the value from the specified analog pin. */
```

TIME

```
delay(time_ms);
```

```
/* Pauses the program for the amount of time (in milliseconds). */
```

```
delayMicroseconds(time_us);
```

```
/* Pauses the program for the amount of time (in microseconds). */
```

```
millis();
```

```
/* Returns the number of milliseconds since the board began running the current program. max: 4,294,967,295 */
```

```
micros();
```

```
/* Returns the number of microseconds since the board began running the current program. max: 4,294,967,295 */
```

```
/* The 'for' statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. */
```

ATTINY85 PINS

Pins 0 - 4 are all general purpose I/O pins (GPIO).

Both `digitalWrite()` and `digitalRead()` can be used with any of these pins.

Pins 0 & 1 are setup for PWM output using `analogWrite()`.

Pins A1, A2, A3 are setup for reading sensor input with `analogRead()`.

DATA TYPES

```
void // nothing is returned
boolean // 0, 1, false, true
char // 8 bits: ASCII character
byte // 8 bits: 0 to 255
int // 16 bits: -32,768 to 32,767
unsigned int // 16 bits (unsigned)
long /* 32 bits: -2,147,483,648 to 2,147,483,647 */
unsigned long // 32 bits (unsigned)
float // 32 bits, signed decimal
```

CONSTANTS

```
HIGH \ LOW
INPUT \ OUTPUT \ INPUT_PULLUP
true \ false
```

