

# WPI C Primer Sheet

## Comments

//Single Line Comment  
/\*Multiline comment\*/

## if Statements

```
if(conditional)  
{  
    //Do something here  
}  
else if(conditional) //Optional section  
{  
    //Do something here  
}  
else //Optional section  
{  
    //Do something here  
}
```

## while Loops

```
while(conditional)  
{  
    //Do something here  
}
```

## for Loops

```
for(initialization;  
    conditional;  
    do this after each iteration)  
{  
    do something here  
}
```

## Functions

```
Return FunctionName(Input Variable list)  
{  
    //Body  
    return Return Value;  
}
```

*Return* is the return data type

*Input Variable list* is the list of Variables denoted by "datatype variablename" pairs

## Data Types

Type	Min Value	Max Value	Memory Size	Speed for basic operations
char	-128	128	1 byte	Very Fast
int	-32768	32767	2 bytes	Fast
long	-2147483648	2147483647	4 bytes	Average
float	Negatively Virtually infinite	Virtually infinite	4 bytes	Painfully slow

## WPI Operations

**void Drive**(*int speed*, *int direction*)

Basic drive operation

**void Motor**(*int PWMport*, *int speed*)

Drive motor at *PWMport* at speed *speed* between -128 and 127

**void Motors**(*int leftSpeed*, *int rightSpeed*)

Drive left motors at *leftSpeed* and drive right motors at *rightSpeed* where both speeds are between -128 and 127

**void TwoWheelDrive**(*int leftMotor*, *int rightMotor*)

Setup motor at port *leftMotor* to be the left motor and motor at port *rightMotor* to be the right motor in a two wheel drive robot

**void FourWheelDrive**(*int leftMotor*, *int frontLeftMotor*, *int rightMotor*, *int frontRightMotor*);

Setup motor at port *leftMotor* to be the rear left motor, and motor at port *frontLeftMotor* to be the front left motor, etc. in a four wheel drive robot

**int PWMIn**(*int port*)

Returns the value of port *port*

# WPI C Primer Sheet

## Common Functions

**void WPILInitialize(void);**

-Library required function, run first before anything else

**void TwoWheelDrive(int leftMotor, int rightMotor);**

-Setup a two wheel drive robot with the left motor plugged into port **leftMotor** and the right motor plugged into port **rightMotor**

**void FourWheelDrive(int leftMotor, int frontLeftMotor, int rightMotor, int frontRightMotor);**

-Setup a four wheel drive robot (see **TwoWheelDrive**)

**void Wait(int ms);**

-Used for sleeping for a specified number (**ms**) of milliseconds

**void Motor(int pwmPort, int speed);**

-Control a specific motor specified by **pwmPort** using speed defined by **speed**

**void Motors(int leftSpeed, int rightSpeed);**

-Drive the left side drive motors at **leftSpeed** and the right side drive motors at **rightSpeed**

**void Drive(int speed, int direction);**

-Drive the robot with a forward speed **speed** turning in direction specified by **direction**

**int PWMIn(int port);**

-Get the value from the radio on port **port**

**int DebugPrintf(rom const char \*format, ...);**

-Print some debug statements, only every 100ms

**int printf(rom const char \*format, ...);**

-Print some debug statements, no time restrictions on printing

**int Get\_Analog\_Value(rc\_ana\_inXX)**

-Gets analog value at port **XX** (where **XX** is between 01 and 08)

### Comparators

**==** equality

**<=** less than or equal

**<** less than

**>** greater than

**>=** greater than or equal

**!=** not equal

### Conditional Modifiers

**&&** Logical AND

**||** Logical OR

**!** Logical NOT

**()** Controls order of operation

### Other Useful Functions

**#define upplevel 9**

**#define IRTToInches(x) (((6787.0 / ((float)(x) - 3.0)) - 4.0) / 2.54)**

Automatically replaces **upplevel** or **IRTToInches()** in code with whatever follows

### Joystick to Radio Assignments

1 Right X-Axis

2 Right Y-Axis

3 Left Y-Axis (Not Spring Loaded!)

4 Left X-Axis