EasyC Sample Code Annotated

```
void main ( void )                          \\This starts the program

{                                           \\Anything between the main section's open
                                            \\bracket and while loop will only run once.

                                            \\Although there is nothing here, if there
                                            \\were, it would only run once at the start


        while ( 2<3 )                       \\The while loop will run over and over until
        {                                   \\the statement in parentheses is false (this
                                            \\example will run forever (until robot is
                                            \\turned off) since 2 is always less than 3.

        JoystickToMotor (  1 , 1 , 5 , 0 ) ;    \\This command will use the joystick to
                                                 \\power an extra  motor (not the wheels).

        JoystickToServo ( 1 , 2 , 6 , 0 ) ;     \\This command will use the joystick to
                                                 \\power a servo

        Arcade2 ( 1 , 3 , 4 , 2 , 9 , 1 , 0 ) ; \\This command will use the joystick to
                                                 \\control the two wheel motors using
                                                 \\arcade drive (as opposed to tank drive).

        }                                   \\This close bracket signifies the end of the
                                            \\while loop. If the while loop it still true (2 is
                                            \\still less than three), it will go back to the
                                            \\beginning of the while loop and run it
                                            \\again. If the while loop statement is false,
                                            \\then the robot will follow any commands
                                            \\that come after this bracket.

}                                           \\This close bracket signifies the end of the
                                            \\entire program and the robot will stop
```

JoystickToMotor (joystick number, channel, motor number, inversion)

- Joystick number - we only use one joystick, so this will always be 1
- Channel - see joystick diagram for channel numbers
- Motor number - which port is the motor plugged into on the cortex?
- Inversion - 0 if no, 1 if yes. If the motor is running backwards to what you want, switch the inversion number.

JoystickToMotor (1, 1, 5, 0) Means channel 1 (right analog joystick left/right) will control the motor plugged into port #5 on the cortex.

JoystickToServo (joystick number, channel, servo number, inversion)

- Joystick number - we only use one joystick, so this will always be 1
- Channel - see joystick diagram for channel numbers
- Servo number - which port is the servo plugged into on the cortex?
- Inversion - 0 if no, 1 if yes. If the servo is running backwards to what you want, switch the inversion number.

JoystickToServo (1, 2, 6, 0) means channel 2 (right analog joystick up/down) will control the servo plugged into port #6 on the cortex

Arcade2 (joystick number, channel 1, channel 2, motor 1, motor 2, inversion 1, inversion 2) Arcade uses one joystick knobs to control 2 motors.

- Joystick number - we only use one joystick, so this will always be 1
- Channel 1 - which channel controls forward/backward movement
- Channel 2 - which channel controls left/right movement
- Motor 1 - which port is the first motor plugged into on the cortex
- Motor 2 - which port is the second motor plugged into on the cortex
- Inversion 1 - 0 if no, 1 if yes. If motor 1 is running backwards to what you want, switch the inversion number.
- Inversion 2 - 0 if no, 1 if yes. If motor 2 is running backwards to what you want, switch the inversion number.

Arcade2 (1, 3, 4, 2, 9, 1, 0) means that the left analog joystick will control the driving motors. Up/Down makes the robot move forward/backward. Left/Right makes the robot rotate left/right.