

The DEFAULT Code

TEAM

and

C Programming

4 and 11 November 2006

FORD  
FIRST

Mr. Bob Koehl

[ace4bert@sbcglobal.net](mailto:ace4bert@sbcglobal.net)

[www.teamfordfirst.org/Events](http://www.teamfordfirst.org/Events)

# Preparing to program a FIRST Robot in C

- **Describe the software needed for robot programming**
- **What is the DEFAULT Program?**
- Learn the default names of items connected to the robot controller and operator interface
- Learn critical C Program Keywords, Operators, and Functions.
- Change the DEFAULT Program
- Use the Dashboard Viewer to diagnose programs.

# MPLAB Software

- Software comes with the Kit of Parts
- MICROCHIP MPLAB Programming Management Software VERSION 7.2 (Used to EDIT programs)
- ([ww1.microchip.com/downloads/en/DeviceDoc/MP720\\_Full.zip](http://ww1.microchip.com/downloads/en/DeviceDoc/MP720_Full.zip)) NEWER Versions WILL NOT WORK properly with the robot controller.
- MICROCHIP C18 C Language Compiler (Converts C to a “robot understandable” file.)
- MUST be version 2.40. (ONLY available in last year’s Kit of Parts). NEWER VERSIONS will not work and could not be located on the MICROCHIP website.

# IFI Robotics Software

- Software comes with the Kit of Parts
- Innovation FIRST, Inc. is the supplier
- IFI Loader Version 1.1.0 used to load the “robot understandable” file into the Robot Controller.
- [www.ifirobotics.com/rc.shtml](http://www.ifirobotics.com/rc.shtml) is the link to this and many other software tools for the robot control system. ([pdf version](#))

# DEFAULT Program

- Innovation FIRST, Inc. is the supplier
- [www.ifirobotics.com/docs/frc-code-2-28-2006.zip](http://www.ifirobotics.com/docs/frc-code-2-28-2006.zip) is the link to the group of files that makes up the DEFAULT Program.
- When “unzipped” there are multiple file types that are used to program the robot using MPLAB and the C18 Compiler.

# DEFAULT Program file list

## **user\_routines\_fast.c**

user\_SerialDrv.c

ifi\_startup.c

ifi\_utilities.c

## **user\_routines.c**

main.c

## **user\_routines.h**

user\_SerialDrv.h

ifi\_default.h

ifi\_aliases.h

delays.h

ifi\_utilities.h

- Multiple **.c** files are the actual C Program files that are used to create the “robot understandable”

### **.hex** file

- Multiple **.h** files are the header files used by the C Program files to get definitions or values

- **These files are usually changed by teams to suit their robot or to add autonomous modes.**

# DEFAULT Program file list

## FRC\_Default.hex

clearFiles.bat

FRC\_MASTER\_V12.BIN

FRC\_alltimers\_8722.lib

FRC\_alltimers\_8520.lib

FRC\_library\_8722.lib

FRC\_library\_8520.lib

18f8520.lkr

18f8722.lkr

FrcCode.mcp

FrcCode.mcs

FrcCode.mcw

- FRC\_Default.hex is the “robot understandable” version of the DEFAULT Code that we will discuss today.
- FrcCode.hex will be created if the program files are changed and then “compiled” to make them “robot understandable”

# DEFAULT Program file types

- FRC\_Default.**hex** is the “robot understandable” version of the DEFAULT Code that we will discuss today.
- Multiple **.c** files are the actual C Program files that are used to create the .hex file
- Multiple **.h** files are the header files used by the C Program files to get definitions or values

# DEFAULT Program Part of file:

## **user\_routines.c**

// This maps the joystick axes to specific PWM outputs.

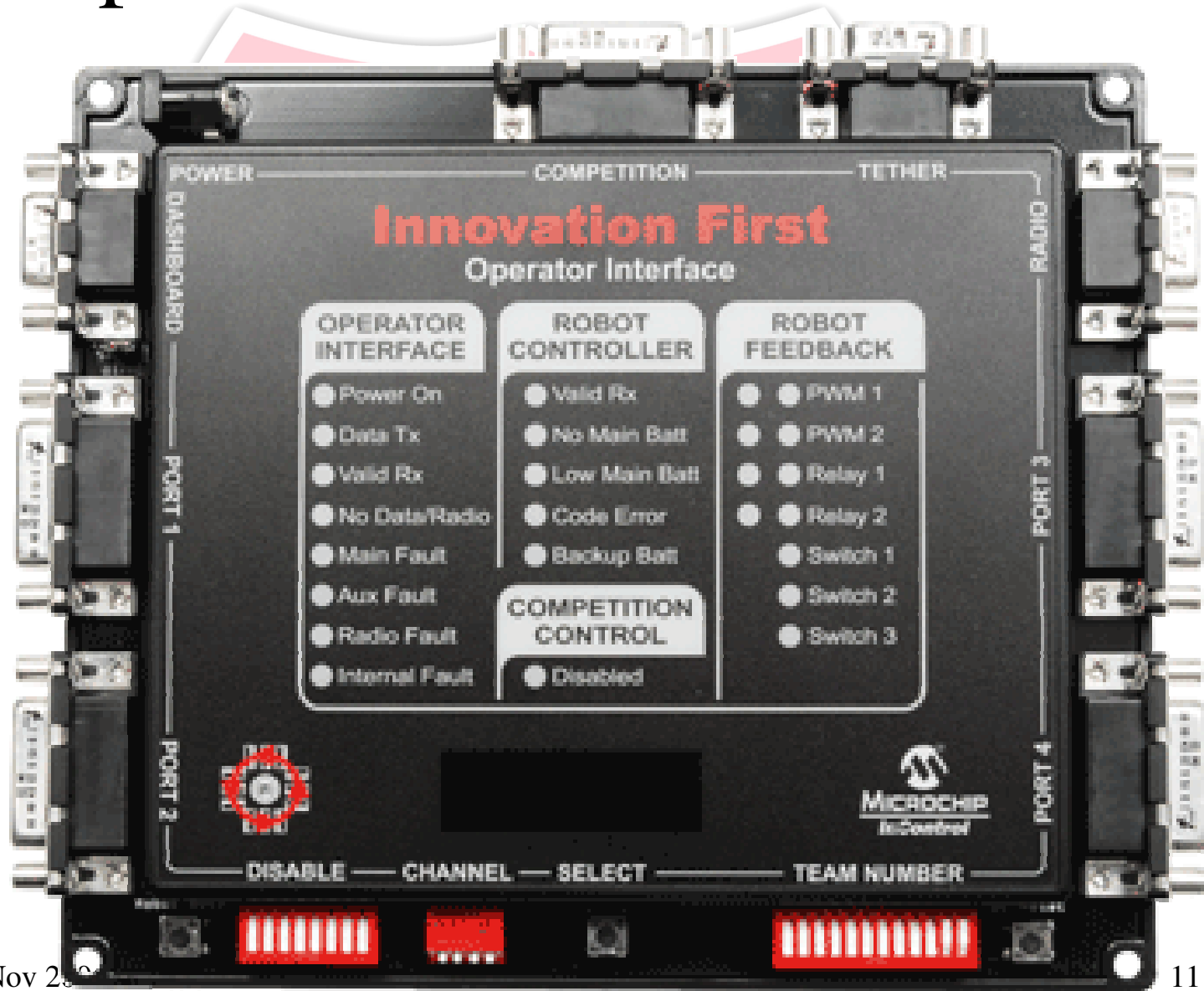
```
pwm01 = p1_y;           pwm05 = p1_x;
pwm02 = p2_y;           pwm06 = p2_x;
pwm03 = p3_y;           pwm07 = p3_x;
pwm04 = p4_y;           pwm08 = p4_x;
pwm09 = p1_wheel;      pwm11 = p3_wheel;
pwm10 = p2_wheel;      pwm12 = p4_wheel;
```

1. THIS IS PART OF A C PROGRAM!!! ([file](#))
2. What does this mean to the ROBOT??

# Preparing to program a FIRST Robot in C

- Describe the software needed for robot programming
- What is the DEFAULT Program?
- **Learn the default names of items connected to the robot controller and operator interface**
- Learn critical C Program Keywords, Operators, and Functions.
- Change the DEFAULT Program
- Use the Dashboard Viewer to diagnose programs.

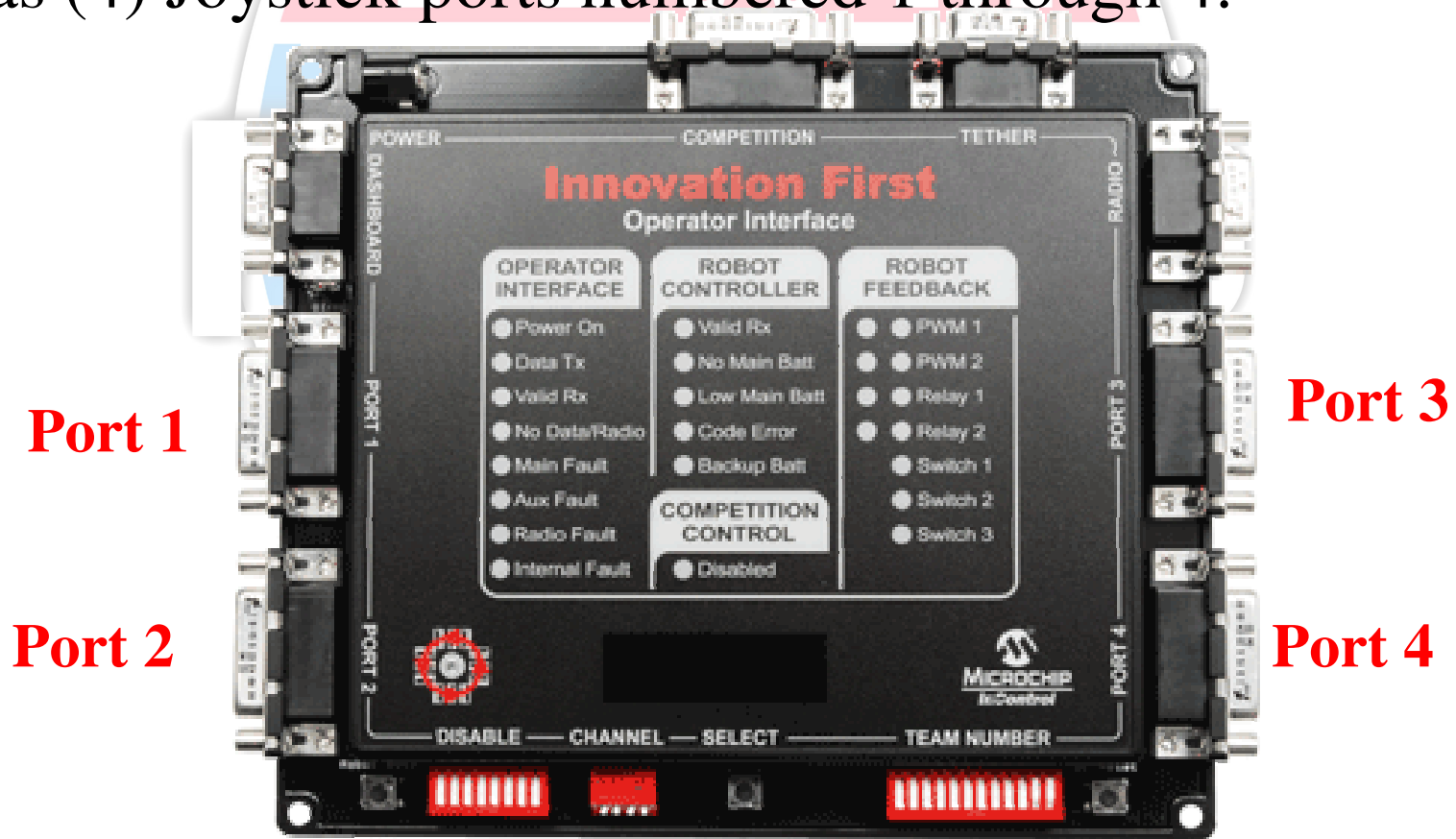
# Operator Interface Board



# Operator Interface Description

Used by Operators to run the robot.

Has (4) Joystick ports numbered 1 through 4.



# Operator Interface **Port 1 NAMES**

Each joystick has:

y-axis (forward and reverse) called **p1\_y**

x-axis (left and right) called **p1\_x**

wheel (either a thumbwheel or a thumb controlled “hat switch”) called **p1\_wheel**

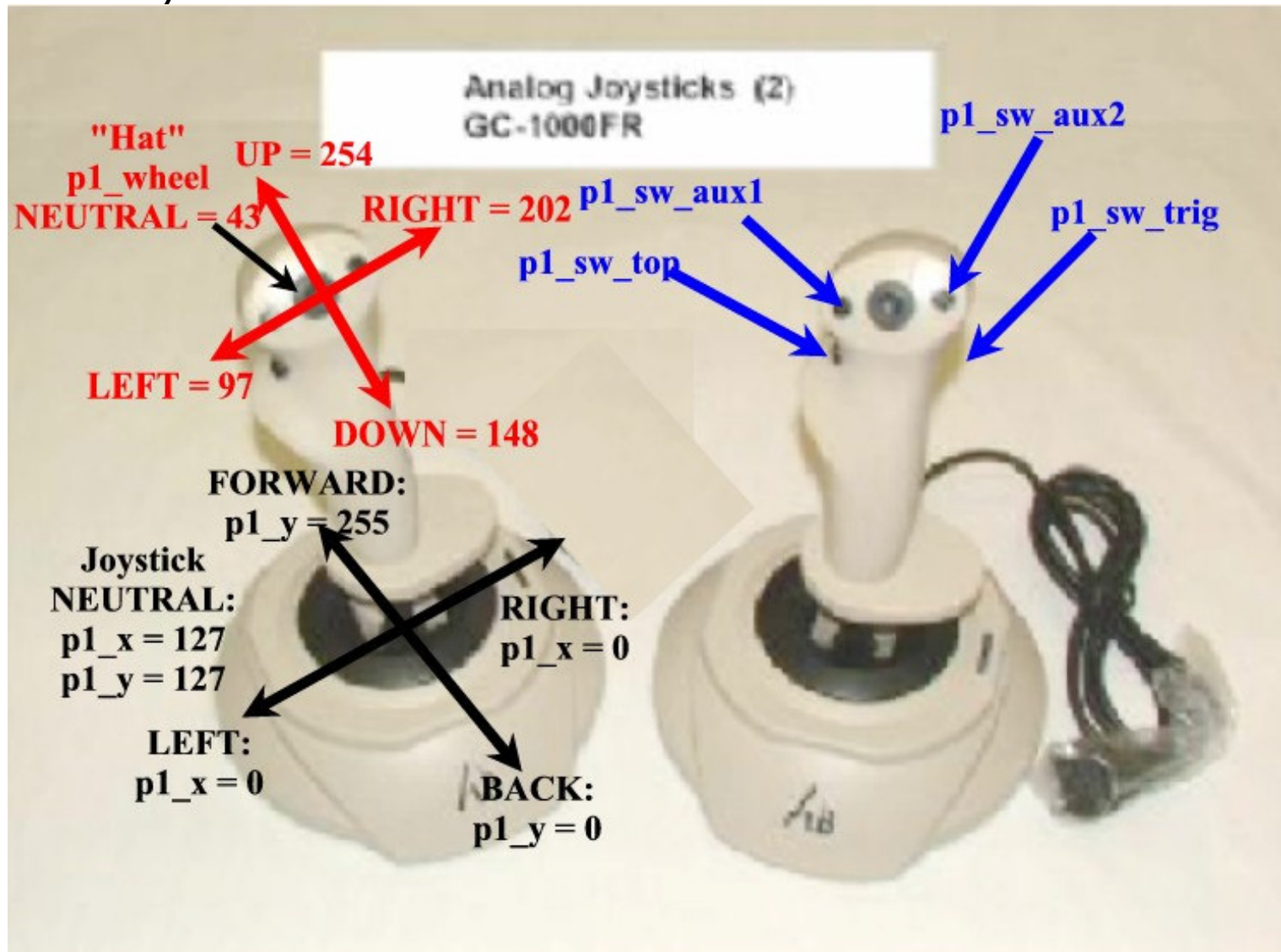
trigger called **p1\_sw\_trig**

top switch called **p1\_sw\_top**

(2) thumb switches called **p1\_sw\_aux1 & p1\_sw\_aux2**

Teams can also add an additional variable switch. called **p1\_aux**

# Joystick on **Port 1 NAMES**



# Operator Interface Ports 2 - 4

If Port 1 y-axis (forward and reverse) is called **p1\_y**  
then port 2 y-axis will be called **p2\_y**

If Port 1 x-axis (left and right) is called **p1\_x**  
then port 2 x-axis will be called **p2\_x**

If Port 1 wheel is called **p1\_wheel**  
then port 4 wheel will be called **p4\_wheel**

If port 1 trigger is called **p1\_sw\_trig**  
then port 3 trigger is called **p3\_sw\_trig**

Learning the “NAMES” of one port applies to all ports.

# Operator Interface Lights

Display information to the Operator through the lights

Robot Feedback lights are named:

**Pwm1\_red**

**Pwm1\_green**

**Pwm2\_red**

**Pwm2\_green**

**Relay1\_red**

**Relay1\_green**

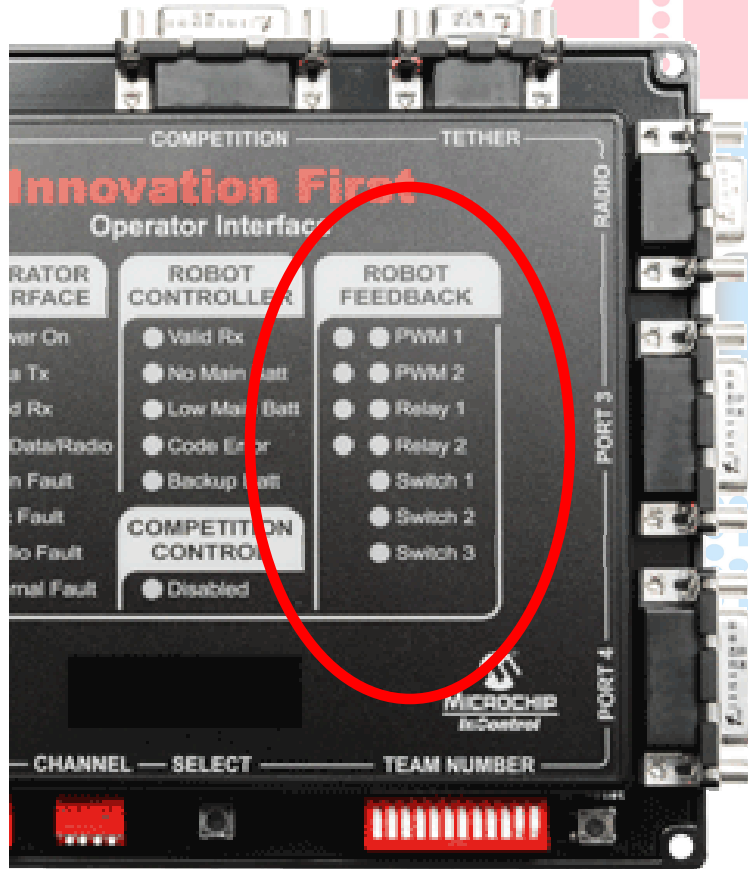
**Relay2\_red**

**Relay2\_green**

**Switch1\_LED**

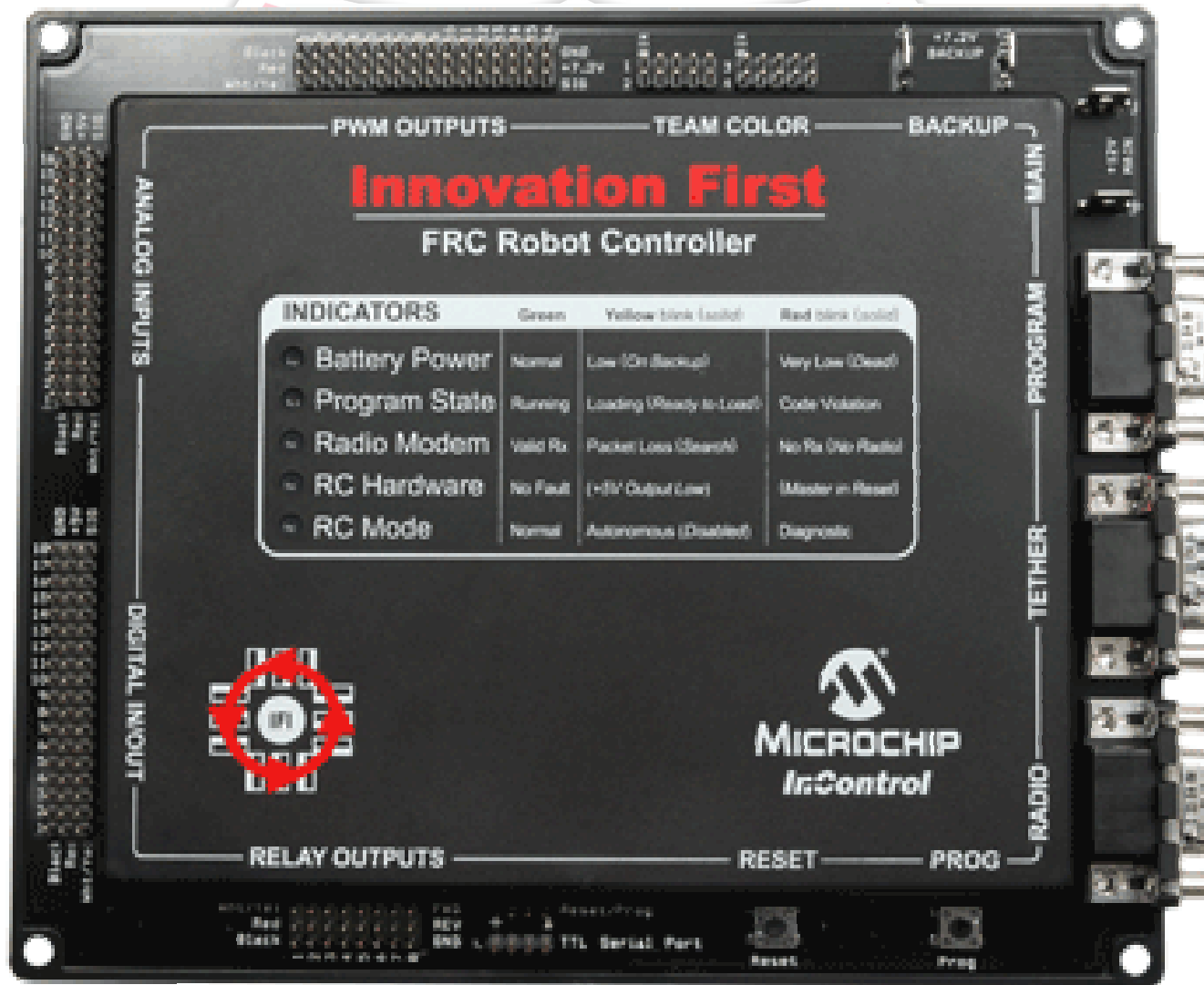
**Switch2\_LED**

**Switch3\_LED**



Lights are ON when set to a value of 1 and OFF when set to 0.

# Robot Controller



4 and 11 Nov 2006

17

# Robot Controller pwm outputs

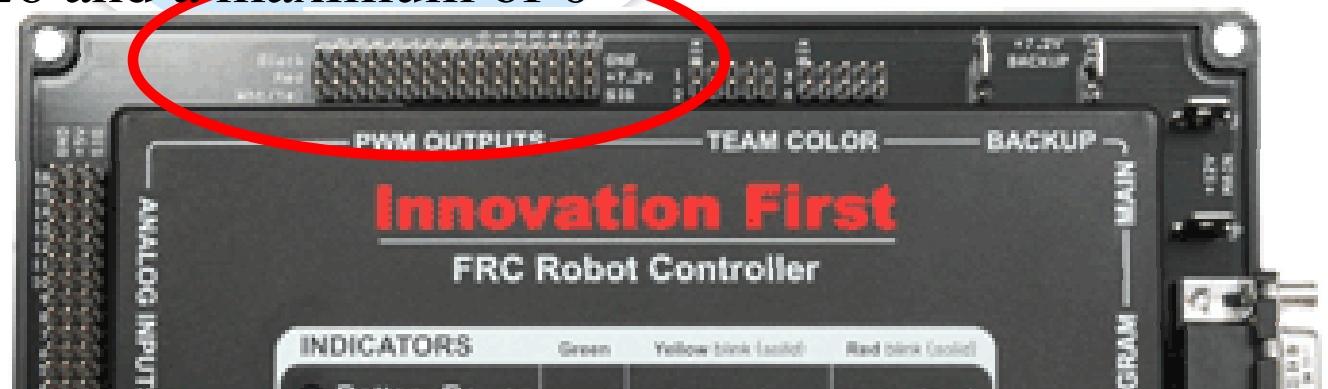
pwm outputs are used to control motors that are connected to VICTOR servo motor controllers at variable speeds both forward and reverse.

The outputs are named **pwm01**, **pwm02**, through **pwm16**

A pwm value set to 127 tells the motor to STOP

A pwm value greater than 127 means “GO FORWARD” with a variable speed between 128 and a maximum of 255

A pwm value less than 127 means “GO REVERSE” with a speed between 126 and a maximum of 0



4 and 11 Nov 2006

## from user\_routines.c

Remember this C Program statement???

```
pwm01 = p1_y;
```

WHAT DOES THIS MEAN to the ROBOT?

Lets' talk our way through it.

## user\_routines.c

```
pwm01 = p1_y;
```

**p1\_y** is the Joystick on Port 1 y axis value between 0 and 255 with 127 as its neutral value.

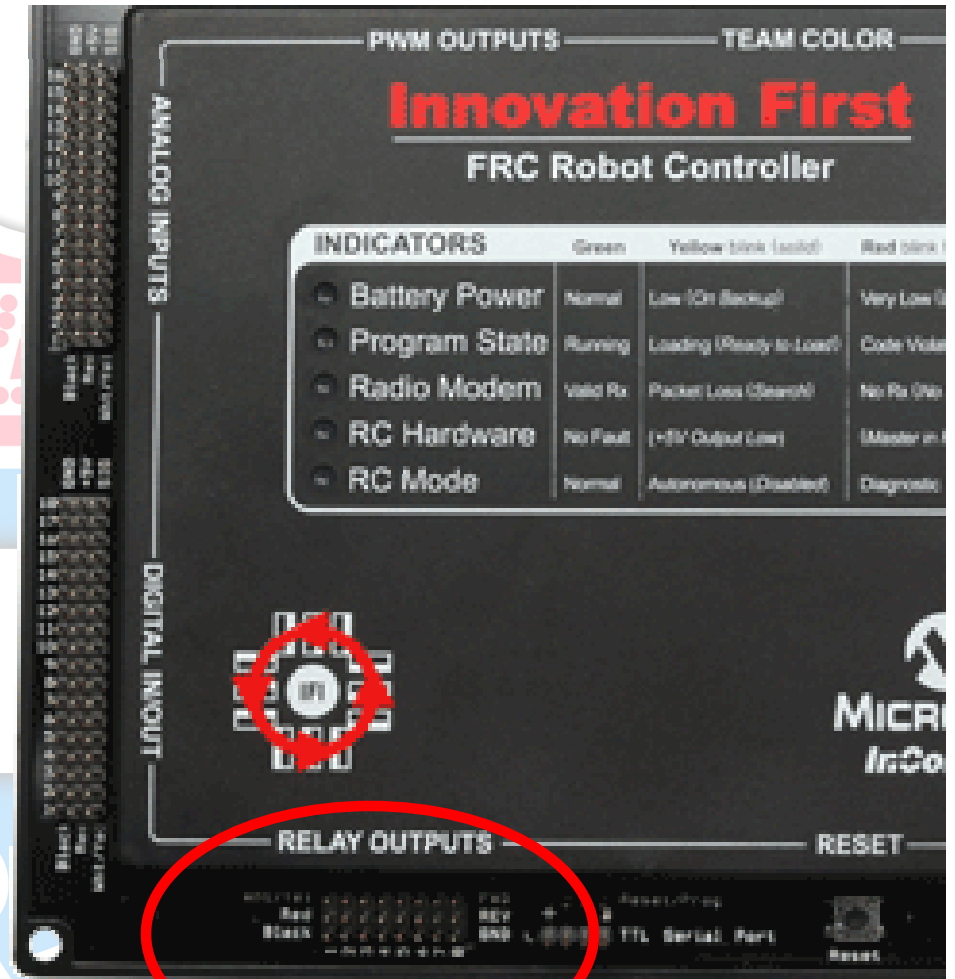
**pwm01** is set to the same value as the Joystick y axis so it will also have a value between 0 and 255, based on the Joystick.

Result: The motor connected to pwm01 will run in the same direction and speed as commanded by the Joystick y axis value. 0 is full speed REVERSE, 255 is full speed FORWARD, and 127 is STOP.

# Robot Controller relay outputs

Relay outputs are used to control devices that run a constant speed or can only be turned ON or OFF.

The outputs are named **relay1\_fwd**, **relay1\_rev**, through **relay8\_fwd** & **relay8\_rev**.



When the relay8\_fwd is turned ON, the motor, valve, or light that is connected to the relay8 SPIKE will function.

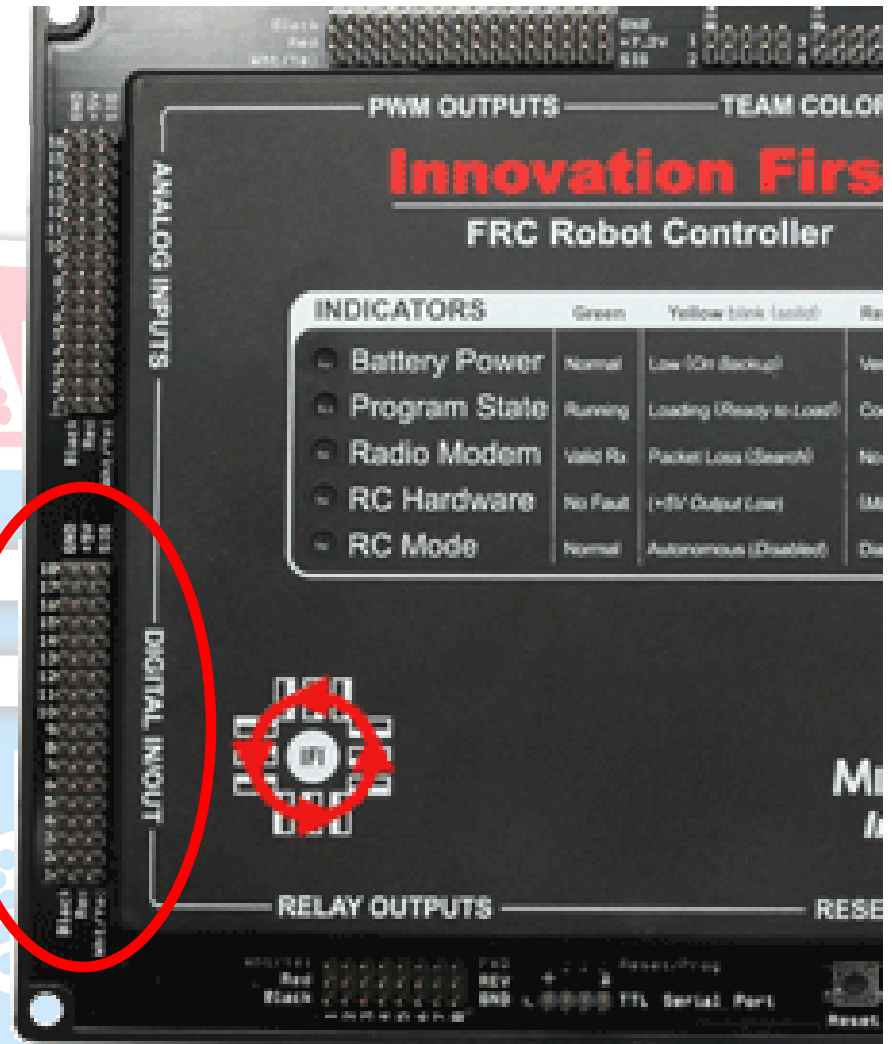
# Robot Controller digital inputs

Digital inputs that are wired to the robot are devices that can either be ON or OFF.

Such devices are limit switches, photocells, and toggle switches.

The inputs are named **rc\_dig\_in01**, **rc\_dig\_in02** through **rc\_dig\_in18**.

The C Program can look at the digital input and help control the robot.



# DEFAULT Program Part of file:

**user\_routines.c**

// This default code maps the joystick buttons to specific relay outputs.

```
relay3_fwd = p3_sw_trig;
```

```
relay3_rev = p3_sw_top;
```

What do these mean to the robot??

# DEFAULT Program Part of file:

**user\_routines.c**

```
relay3_fwd = p3_sw_trig;
```

**p3\_sw\_trig** is the trigger on Joystick 3.

When the trigger is activated by the operator,  
**relay3\_fwd** will turn ON.

When the trigger is NOT activated, **relay3\_fwd** will be turned OFF.

If relay3\_fwd was connected to a motor, the motor would RUN while the trigger was pulled and STOP when the trigger was released.

# DEFAULT Program Part of file:

**user\_routines.c**

```
relay3_rev = p3_sw_top;
```

Talk your way through it:

What is **p3\_sw\_top**?

When happens when the **top** is activated?

When happens when the **top** is NOT activated?

If **relay3\_rev** was connected to a motor, the motor would RUN while the top was pressed and STOP when the top was released.

# Default Names



- Operator Interface names begin with the port number where the device is plugged in; namely p1 through p4 for port 1 through port 4.
- Robot Controller devices are named by the “bank of connectors” where the devices are plugged in; namely pwm outputs, relay outputs, digital in/outs , and analog inputs.

# Name the connections to the Robot Controller (rc)

There are four kinds of connections to the robot controller:

1. Outputs labeled pwm.
2. Outputs labeled relay.
3. Inputs labeled digital.
4. Inputs labeled analog.

# pwm output connections

- There are 16 pwm output connections.
- They are named **pwm01**, **pwm02**, through **pwm16**.
- They are used to give motor speed (stop through full speed) and direction (forward or reverse).
- They are connected either to Victor speed controllers or directly to Servo Motors

# Relay output connections

- There are 8 relay output connections.
- They are named relay1, relay2, through relay8.
- Each relay output has a forward and a reverse.
- Forward outputs are named **relay1\_fwd**, **relay2\_fwd**, through **relay8\_fwd**.
- Reverse outputs are named **relay1\_rev**, **relay2\_rev**, through **relay8\_rev**.
- They are connected Spike relays which can run either motors or valves.

# Switch digital input connections

- There are 18 switch input connections.
- They are named **rc\_dig\_in01**, **rc\_dig\_in02**, through **rc\_dig\_in18**.
- The robot controller may have many different types of digital switches connected including:
  - Buttons or Toggle switches
  - Limit switches or Air pressure switches
  - Encoders or Photo sensitive switches
  - Any switch which can be either ON or OFF

# Switch analog input connections

- There are 16 analog input connections.
- They are named **rc\_ana\_in01**, **rc\_ana\_in02**, through **rc\_ana\_in16**.
- The robot controller may have many different types of analog switches connected including:
  - Potentiometers
  - Gyros or Accelerometers
  - Pressure or Force Transducers
  - Any device which provides multiple values

