







**Team Ford First**   

---

## Adding to the DEFAULT Program and Sensors

by  
**Bob Koehl**  
 Team Ford FIRST  
 Presented by  
 Marvin Kraska  
 mmee@wowway.com




**Team Ford First**   

---

## Robot Control System Data

Download from the IFIRobotics website:  
[www.ifirobotics.com](http://www.ifirobotics.com)  
[Robot Control System Reference Guides](#)  
[Robot Controller Documentation](#)  
[Operator Interface Documentation](#)

November 2005 2




**Team Ford First**   

---

## Robot Software System Data

Download from the IFIRobotics website:  
[www.ifirobotics.com](http://www.ifirobotics.com)  
 Latest version of:  
[DEFAULT Program \(zip\)](#) the robot program  
 ([DEFAULT Program Reference guide](#))  
[Program Loader \(zip\)](#) to reprogram the robot  
 ([Programming Reference Guide](#) pg 22)  
[Dashboard Viewer \(zip\)](#) to troubleshoot the robot  
 program ([DashBoard Viewer Specifications](#))

November 2005 3




**Team Ford First**   

---

## Programming Software

- MicroCHIP provides a "project manager" program called [MPLAB\\_IDE](#) which can be downloaded from the web.
- Program files are edited, saved, and compiled in the computer using MPLAB.
- The C18 Programming Language Compiler is provided to each team with the Kit of Parts.

November 2005 4




**Team Ford First**   

---

## Setup to Change DEFAULT

- "UNZIP" the DEFAULT Code
- Start MPLAB\_IDE Program
- Create New Project from the DEFAULT
  - Include the date in the new Project name.
  - Copy the files from the old to the new Project.
  - Do this EVERY time a major change is made.
  - This keeps a file of the old version that worked.

November 2005 5


**Team Ford First**   

---

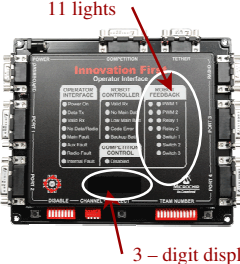
## Default Program Overview – Robot/Human Interface

- Robot Interface is mapping control program to inputs and outputs
- Substantial I/O already provided, try to preserve compatibility, delete/replace new functions
- Human interface is mapping driver intentions to control program
- Choose sensible defaults, test with multiple people

November 2005 6


**Team Ford First** 

## Program the Operator Interface

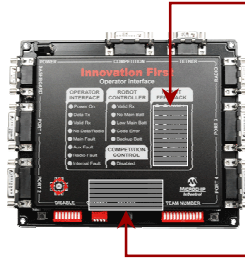


- See `ifi_aliases.h` for the name of each light on the Operator Interface (OI).
- With this information we see the names of each programmable indicator on the OI on the next slide.

November 2005 7

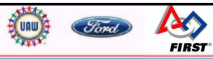
**Team Ford First** 

## OI Display Names



- 11 lights
  - Pwm1\_red Pwm1\_green
  - Pwm2\_red Pwm2\_green
  - Relay1\_red Relay1\_green
  - Relay2\_red Relay2\_green
  - Switch1\_LED
  - Switch2\_LED
  - Switch3\_LED
- 3 - digit display
  - User\_Mode\_byte


November 2005 8

**Team Ford First** 

## Search the DEFAULT Program

- Find in the DEFAULT Program where the OI lights and display are programmed.
- Do a "Project Search" for "Pwm1\_green"
- How would you change what it shows?
- Do a "Project Search" for "User\_Mode\_byte"
- How would you change what it shows?


November 2005 9

**Team Ford First** 

## Create new Program

- Change the 3 digit display so it shows the value being sent to pwm01.
- The program `user_routines.c` contains:
  - `User_Mode_byte = backup_voltage*10;`
  - Let's change it to the Y value of Joystick 1:
  - `User_Mode_byte = p1_y; //this makes the change`


November 2005 10

**Team Ford First** 

## Program Compile and Download

- Make or Build the project in MPLAB
- Fix any errors that are identified.
- Start the program IFILoader
- Select the file just created.
- Place the Robot Control in Program Mode by pressing and holding the Program Button
- Download the new program.




November 2005 11

**Team Ford First** 

## Confirm Program Works

- Robot Controller should automatically leave the Program Mode and begin to Run.
- Press the Mode Select Button on the Operator Interface until it displays the `User_Mode_byte`.
- Should be near 127
- Move Joystick 1 and observe the changes.




November 2005 12

**Team Ford First**   

## Modification Process

- Organization
- Discipline
- Re-use
- Testing
- Revision Control




November 2005 13

**Team Ford First**   

## Modification Process – Organization & Discipline

- Top-down decomposition
- Keep files small (<1pg code)
- Use naming conventions – files & variables
- Bookeeping – header files & block comments, inputs & outputs
- Space is your (brain's) friend




November 2005 14

**Team Ford First**   

## Modification Process – Re-use & Testing

- Design code to be general purpose
- Use simulator to test code before downloading
- Mock up hardware, test on bench
- Use printf()'s to test code after downloading
- After testing, add to your toolkit as a module or library

November 2005 15

**Team Ford First**   

## Library Example




```

/*
 * low_pass
 * Return signed integer filtered value as a constant proportion
 * of old and new arguments.
 * filter constant 'stab' (0.0-0.99), larger values are less filtering (more new)
 * smaller value is more filtering (more old)
 *
 * If filter output would truncate, ramp one bit to make filter converge.
 */
int low_pass(int old, int new, unsigned char fc)
{
    signed char stemp;          /* holds proportion product */
    int temp;

    stemp = HSBYTE((new-old)*fc); /* drive old to new */
    temp = stemp + old;
    if ((stemp!=0) || (new==old)) return(temp); /* filter updated, or converged */
    if (new<old) temp--; else temp++; /* no update, just inc/dec */
    return(temp);
}

```




November 2005 16

**Team Ford First**   

## Modification Process – Revision Control

- Checkpoint code at critical times, simple copy or Zip whole tree.
- Name copies so you can tell at a glance
- Keep a journal/log
- Let the computer help you figure out what changed and why


November 2005 17

**Team Ford First**   

## DEFAULT Sensors

- There are two types of sensors:
- Digital
  - Numerically either 0 or 1
  - Physically either OPEN or CLOSED, or PRESSED or RELEASED
- Includes: Limit Switches, Banner Photo Cells, Air Pressure Switch, Push Buttons on Joysticks, etc. Encoders can be 2 digital input devices.


November 2005 18

**Team Ford First** 

## DEFAULT Sensors

- Analogue sensors indicate the specific position of the device instead of only ON or OFF
  - Value on the Operator Interface is 0 to 255
  - Value in the Robot Controller is 0 to 1023
- Examples include: Joystick x, y, and wheel, Potentiometers (relate to degrees of rotation), Gyro (degrees/second of rotation), etc.


November 2005 19

**Team Ford First** 

## Operator Interface Sensors

- [ifi\\_aliases.h](#) defines NAMES for all of the Inputs possible from the Operator Interface
- Notice that they are named starting with either a p1, p2, p3, or p4 which indicates the desired PORT (1 through 4).
- Notice there is a `_trig`, `_top`, `_aux1`, or `_aux2` for Digital Inputs. Inputs NOT connected == 0.
- Notice that there is an `_x`, `_y`, `_wheel`, or `_aux` for Analog Inputs. Inputs NOT connected == 127.


November 2005 20

**Team Ford First** 

## Robot Controller Sensors

- [ifi\\_aliases.h](#) defines NAMES for all of the Inputs possible from the Robot Controller
- Digital Inputs are defined as `rc_dig_in01` through `rc_dig_in16`. A Digital Input with nothing connected is a 1 == OPEN
- Analogue Inputs are defined as `rc_ana_01` through `rc_ana_16`. The value when read by the controller is between 0 and 1023.


November 2005 21

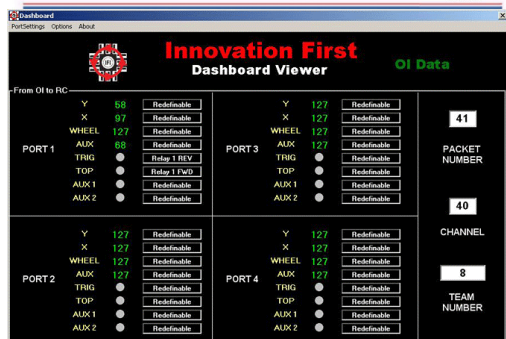
**Team Ford First** 

## Use DashBoard Viewer


- Start the Program `DashBoard_Viewer`
- Connect the Program Cable to the DashBoard Port of the Operator Interface.
- Move the Jumper to show OI Data.
- Move the Jumper back to show RC Data.

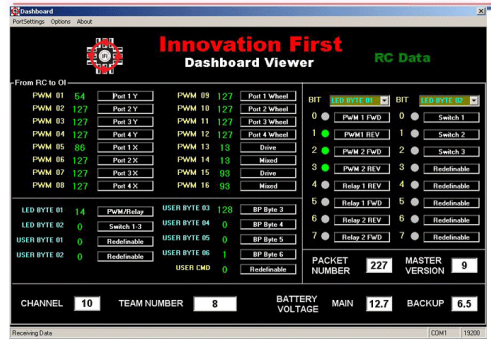
November 2005 22

**Team Ford First** 






November 2005 23

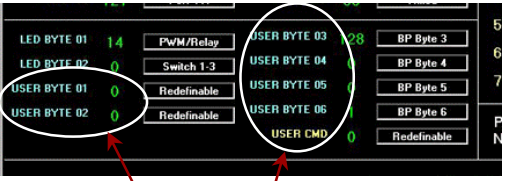
**Team Ford First** 



November 2005 24




**Team Ford First**   

## Display Additional Robot Data



Notice User\_Byte1 through User\_Byte6.  
They can be programmed to display Robot Values.

November 2005 25




**Team Ford First**   

## Add Robot Analogue Input

- Assume the Robot has a potentiometer wired to its Analog Input # 1.
- Use the [Programming Reference Guide](#) (page 16) we see how to read a potentiometer.
- The DashBoard Viewer can only display 8-bits so we need to program:
 

```
sensor1 = Get_Analog_Value( rc_ana_in01 ); //read 10-bits of data
sensor1_8bits = (unsigned char)(sensor1 >> 2);
//shift the 2 least significant bits out is the same as divide by 2
User_Byte1 = sensor1_8bits; //Display on the DashBoard
```




November 2005 26

**Team Ford First**   

## Make the Program Change

- Using MPLAB, add the program lines in user\_routine.c
- "Make" or "Build" the Project.
- Fix any errors and "Make" again.
- Use IFI\_Loader to Download the robot.
- Use DashBoard Viewer to verify the results.




November 2005 27

**Team Ford First**   

## Analog vs Digital

- Who needs five senses?
- Three times is a charm...
- Sanity checking
- Filtering
- deFAULT behaviour




November 2005 28

**Team Ford First**   

## Analog vs Digital – 5 -> 3

- Humans have at most 5 senses(sight, sound,touch,taste & smell) to help them interact with their environment
- Robots have 3 (analog, digital, time) to help them...
- Everything else must converted to one of those 3 dimensions by sensors
- Useful information comes from processing sensor values with computation & logic
- Use a human as an extra sensor if possible




November 2005 29

**Team Ford First**   

## Digital Sensors

- On/Off value
- Limit switches, Banner Optical, encoders, IR sensors
- Mechanical switches bounce, you should debounce (filter) switch before using it
- Level may indicate something
- Transition may indicate something
- Count/Time transitions may be useful


November 2005 30

**Team Ford First**   

## Analog Sensors

- Continuous values converted as discrete numbers.
- Sampling time vs response time
- Voltage, current, position, temperature, distance, etc,
- Past history may be useful, time varying
- Filtering may help




November 2005 31

**Team Ford First**   

## Analog vs Digital - deFAULT

- Bad things will happen, admit it, deal with it
- First, do no harm...
- Wake up and smell the coffee – test on power up or during known conditions.
- Warn human something is wrong
- Manage fault – limited operation, substitute values, have backup strategy




November 2005 32

**Team Ford First**   

## Driver vs Autonomous

- Human in the loop...
- Simulated human inputs
- Decisions, decisions, decisions
- Does anybody know what time it is?
- Where am I?




November 2005 33

**Team Ford First**   

## Driver vs Autonomous - HIL

- Human is critical part of the robot control system
- Humans automatically compensate for a lot of bad things...
- Purpose of human is to generate p1\_x, p1\_y, etc. to accomplish goal
- Add to human senses if possible, but be careful of information overload




November 2005 34

**Team Ford First**   

## Driver vs Autonomous – Simulated Human

- Robot program now has to simulate human interaction
- Input, Process, Output – and keep track of history
- Could work with pwm01 and relay03, but why not generate equivalent p1\_x, p1\_y, etc?




November 2005 35

**Team Ford First**   

## Driver vs Autonomous – Decisions

- Know what to do and when
- Must sequence thru steps in specific order
- Can modify behavior based on sensor or inferred information
- Open loop vs Closed loop
- Logic is only as good as information it works with




November 2005 36

**Team Ford First**   

### Driver vs Autonomous – Timing

- RC is very good at keeping track of time
- Simple playback of p1\_x vs time, etc
- Smarter playback with sensor/logic
- Consider resolution, memory space, and mechanical limits

November 2005 37

**Team Ford First**   

### Driver vs Autonomous – Position

- RC has no direct knowledge of where it is
- Dead-reckoning is utilizing information about what should be happening to infer location
- Improve location information by counting revolutions of wheels, etc.
- Improve location information by other senses – touch, sight
- Improve location information by observing field features




November 2005 38

**Team Ford First**   

### Murphy’s Law - Best plans...

- Make it work all the time, not just sometime
- Be an Optimist in design, a Pessimist in coding
- Keep It Simple, Stupid!!!
- Get basic functions working, then add features & test.
- Always keep a working copy
- “Test like you fly, fly what you test”
- Murphy was an optimist...




November 2005 39

**Team Ford First**   

### Murphy’s Law – HW/SW Failures

- Wires do come loose – know how to debug
- Software is part of the signal chain - GIGO
- Divide and conquer
- Substitute signals to test
- The best tool is your brain




November 2005 40

**Team Ford First**   

### Murphy’s Law – Cockpit Failures

- Humans are not perfect, use the RC/OI to help them
- Simpler interfaces mean less chance for confusion
- Pre-flight check list
- Label everything – last minute substitutions are possible.




November 2005 41

**Team Ford First**   

### Murphy’s Law – Energy Management/Heat of battle

- Pre-flight checklist must include battery state
- Main breaker should never open
- Other circuit breakers may open (at the worst possible time)
- Motors overheat – RC can manage
- Leave it all on the field.




November 2005 42

**Team Ford First**   

## Summary

- Today we learned where to find the DEFAULT Program, Programming and Hardware documentation, DashBoard Viewer, and the Program Loader.
- HINT: [www.ifirobotics.com](http://www.ifirobotics.com)




November 2005 43

**Team Ford First**   

## Summary

- Today we have learned how to find the DEFAULT NAMES of every connection to the Robot Controller and to the Operator Interface.
- HINT: ifi\_aliases.h of the DEFAULT Program
- Also defined in the OI Reference Guide (oi-ref-guide-2-21-05.pdf) and the RC Reference Guide (rc-ref-guide-01-31-2005.pdf)




November 2005 44

**Team Ford First**   

## Summary

- Today we learned how to make different versions of our program PROJECT using the "Program Wizard" in the MPLAB program development system.
- We also learned that the C Compiler software is provided to each team with the Kit of Parts.




November 2005 45

**Team Ford First**   

## Summary

- Today we learned that by using the software tools we can add to the DEFAULT Program.
- We demonstrated displaying sensor values on the Operator Interface panel.
- We demonstrated displaying sensor values through the DashBoard Viewer.

November 2005 46

**Team Ford First**   

## Additional References

**Sensor Development**  
Kevin Watson's webpage, detailed, useful info about Sensors and the RC (read the FAQ)  
<http://kevin.org/rc/>

**ChiefDelphi Web Forum**  
Lots of discussion (technical & non-technical), speculation & rumors...  
<http://www.chiefdelphi.com/>




**Revision Control**  
Free version control system for Windows systems, visual file differencing program  
<http://www.kitware.com/>, <http://wwwmerge.org/>

**Code Documentation Tools**  
Free code documentation tools, create HTML from your code  
<http://www.doxygen.org/index.html>, <http://www.pms.org/software/ezghal/>

**Component Sources**  
Various sensor components that may be of interest - accelerometers, compass, GPS, serial LCD displays  
<http://www.parallax.com/>, <http://www.sparkfun.com/shop/index.php?shop=1&ware=125153&ware=71&>, <http://www.seesem.com/>

**Email contact:**  
Marin Kraska, **IT&M Electrical Engineering**  
[mkr@norway.com](mailto:mkr@norway.com)

November 2005 47

**Team Ford First**   

## Questions ???

November 2005 48