The Adams Coprocessor Solution

Presented and developed by Team 245
The AdamBots

Why should I use a coprocessor?

For our team, the coprocessor was an easy way to work around the often frustrating restrictions of the Microchip 18F Controllers.

Some benefits of the coprocessor:

- Near infinite memory (variables, floating point)
- Fast processing (no 26.2 ms glass ceiling)
- Easier programming (No weird versions of C)

How could I use these benefits?

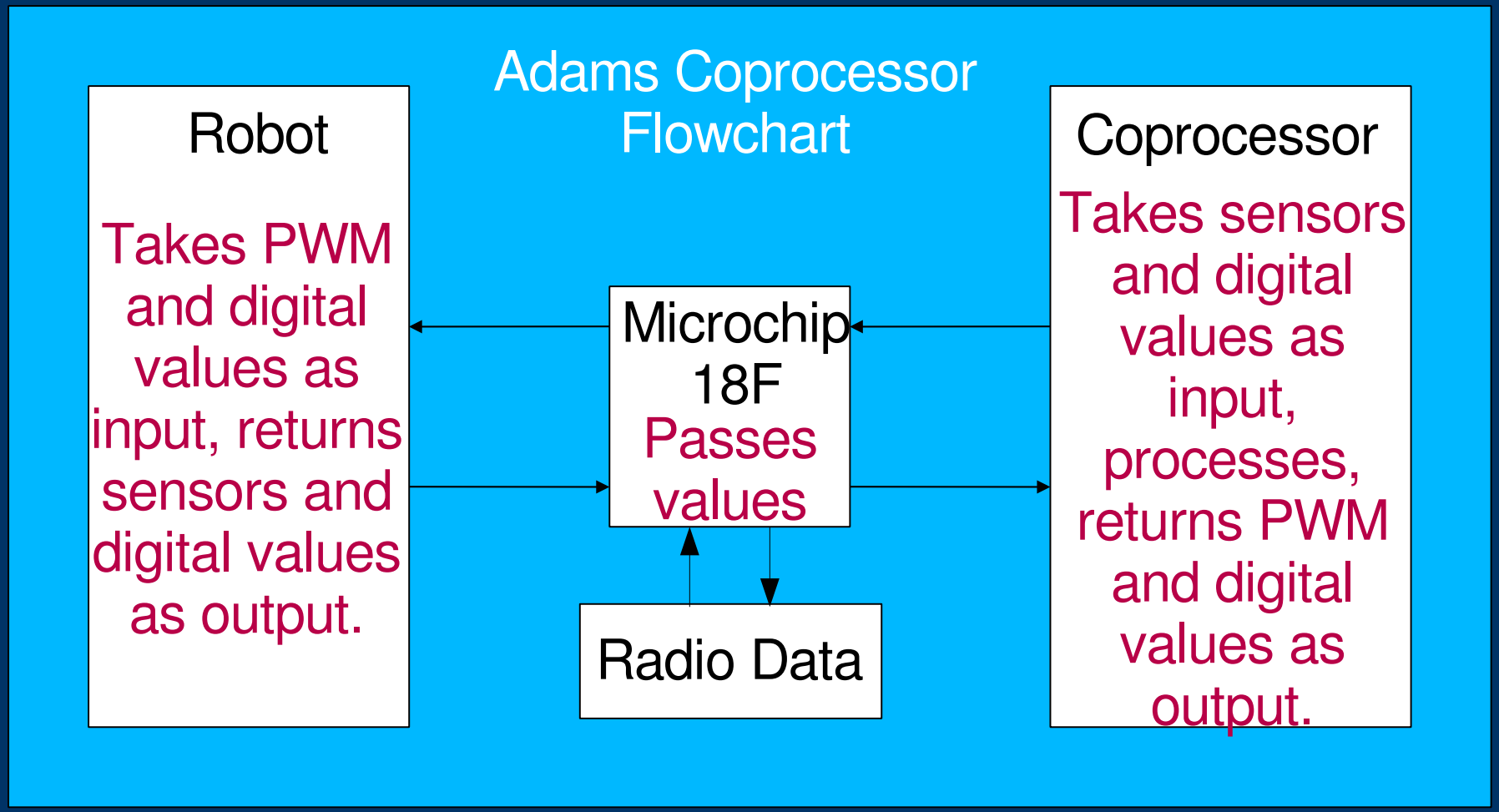
Some specific examples of possible applications:

- All sensors (cameras, gyroscopes)
- Proportional Integral Derivative (PID) Control
- Autonomous – Real time sensing and feedback
- Autonomous – Easy dead reckoning

So how does it work?

The coprocessor is another computer that communicates to the controller via serial through the programming port. The Microchip 18F still communicates to the robot, but it is really only the coprocessor that does any of the complicated computing or processing.

So how does it work?



Adambots Coprocessor Solution

What we have discussed are only the basics of any coprocessor. The AdamBots have already created a fully working coprocessor, however. It is called the Adambots Coprocessor Solution, or the ACS. In the ACS, the Microchip 18F sends its values (with a C program we have already written) to a small Linux computer running a Python script. This script and its associated modules make up the core of the ACS.

So what do we need?

Your team needs to make its own decisions regarding a couple of things. Here is a loose check list of what you need to obtain to use the ACS:

- Microchip 18F (Included in kit)
- Your own coprocessor (Gumstix or microATX recommended) – A laptop works for development
- Our code - available through Bazaar or download http://www.adambots.com/coprocessor/

Listen to the voice of experience!

Team 245 has experience using this solution, and we have learned a few lessons.

1)Coprocessors need love! - Yes, we have actually fried a coprocessor by being way too harsh to it after a match. These are moderately fragile machines, so be careful with them.

2)Coprocessor, not Panacea! - You cannot expect your coprocessor to make your robot magically better. For example, a robot without a solid drive train will probably never be able to follow any kind of sensor reliably.

In Depth Stuff

Now that you have an overview of the ACS, we will examine the different components in greater detail:

- The Hardware
- Interfacing Sensors
  - The Software –
  - Communication
  - IFI Side
  - Coprocessor Side

The Hardware – It's all about $$$

The ACS ***does*** require the purchase of some hardware (less than $200 total) to be compliant with *FIRST* game rules. You **do not**, however, have to purchase **any** hardware to test the ACS. Any laptop capable of running Linux (that'd be pretty much any laptop) can function as a coprocessor for the ACS.

The Hardware – Shopping List

You will need to purchase a suitable coprocessor that complies with *FIRST* rules. While we have (obviously) not yet read the 2007 official rules, we have no reason to believe any of the following options are illegal.

Gumstix:

Gumstix are miniature computers that run on ARM processors (similar to a PalmPilot). They are the original test platform for the ACS, and worked successfully in 2006.

MicroATX:

Yes, a real computer, no we're not crazy. For those who really want to push the limits (raw image processing from a commercial USB webcam, etc), we believe that a motherboard can indeed survive on a robot. Our team will be attempting this route this year, but we make no guarantees for its stability, security, or success.

Interfacing Sensors – Overview

The ACS's greatest benefit is it's ability to manipulate data, but it can only do that as fast as it can recieve it. You may interface sensors to either the IFI controller or your coprocessor, but remember, by interfacing with the IFI, you are slowing your ability to transfer data and the quality of that data.

One of the motivating factors for the switch from Gumstix to MicroATX was the drastic increase in I/O potential with the MicroATX.

Interfacing Hardware – Coprocessor

Interfacing hardware with the coprocessor is the easiest, fastest, and best way for the ACS to input data.

- Physically connect the device to your coprocessor.
- Modify the ACS code to read and process the input.

- Detailed tutorials for connection and code modification are available from our website: http://adambots.com/coprocessor

Interfacing Hardware – The IFI Side

We advise against interfacing complex devices, especially those with large amounts of data (such as the camera or encoders) with the IFI. Simple sensors, such as limit switches are ideal for connecting to the IFI, as they demand little resources and will not increase data bandwidth or processing excessively.

- Physically connect the device to the IFI controller
- Modify the IFI code to pass input on to the coprocessor (or any other purpose as you like)
- Modify the coprocessor code to accept the new data and process it appropriately.

Communication – The Short Story

The ACS communicates over serial using the programming port on the IFI controller. During normal operation, the IFI and the coprocessor exchange data once every 26.2 ms, alternating sending and receiving as the IFI receives new data.

Since the coprocessor is connected to the programming port, it is capable of flashing the IFI directly; if you have a wireless setup, this means never plugging in a single wire ever again.

IFI Software

The IFI software is very simple, it...

- Reads new radio/sensor data and sends it to the coprocessor
- Reads data packets from the coprocessor and outputs PWM, Relay, and Digital/Analog I/O signals to the robot.
- Does **not** do any processing of any kind, although it could if you for some reason desired that.

Coprocessor Software

The default image of the coprocessor software...

- Is threaded; it is constantly working and updating values. Every 'module', PID, vision tracking, etc runs independently. These modules will process data immediately as it becomes available
- Reads the IFI packets and alerts all threads that new data has arrived; they may or may not do anything with this.
- Sends control packets to the IFI, reading the most recent values from every thread immediately before sending.

Programming With the ACS

All of the ACS (except the IFI code) is written in Python. Python is a newer programming language that (in our humble opinion) is easier and more powerful than C.

Since the coprocessor is a true Linux system, you are free to use your favorite programming language, be it a throwback to PBASIC or Perl, if you're into that sort of thing.

Differences Between Python and C

While you are free to program your entire robot in C, inserting them as modules into the ACS (an ability of Python), we believe that the transition is simple enough to warrant the learning curve.

In the next several slides, we will give you the 50 cent tour of Python and highlight some of the key differences.

Object-oriented Language

import don't include:

In C you include header files, which basically appends the file to your main one, bringing all functions with it.

In Python, you import a module, such as os, and it's functions are objects of the os module, so if you wanted to use the getcwd() function from os, type:

location = os.getcwd()

It MUST be Pretty!

In Python, code structure is done by proper use of whitespaces, tabbing, newlines, etc

```
if awesome == True:
        print “You've won”
        return True
elif exception == True:
        pass
else:
        if alternate == True: return True
        else:
                print “You've failed, sorry”
                return False
```

The End – So sad, isn't it?

More than anything else, we urge you to TRY this. It requires no extensive setup or commitment. Simply download the ACS from our website, flash your IFI, plug in a laptop and try it out. See if you like it.

We'd love to answer any more questions you may have about the ACS. This power point is meant to educate you on the basics of the system and hopefully get you interested. If you would like to know about the ACS in more detail, please ask us about it and try it out for yourself!

http://adambots.com/coprocessor/