Micro-controllers

Applications in Experimental Aircraft

RAA Meeting
Nov 14, 2016

Terry Fisher

About Me

- Retired – software/hardware designer
- Pilot – Cessna 182T C-GYKF, G1000
- Building a Rotorway Exec 162F
- Mentor ECI FIRST Robotics Team
- Have too many hobbies
- May have too many toys...

What can micro-controllers do?

- Digital Inputs
  - can be used to monitor status of systems
  - detect pilot inputs (buttons/switches/etc.)
- Analog Inputs
  - Measure voltages
  - Can often be configured as Digital I/O
- Digital Outputs
  - control relays, LEDs, LCDs, etc.
- Communications
  - Protocol converters between different systems

Why Use Micro-controllers

- Small
- Reliable
- Simple
- Low Power
  - processors can run for years on batteries
- Easy to modify/update functionality
- Inexpensive
- Wide range of sensors / outputs available

Alternatives to micro-controllers

- Buy an off the shelf solution
  - expensive (and not much fun)
- Install a light for every status and switch for every function
  - not very intuitive/user friendly
- Combinational logic (AND, OR gates)
  - requires designing/soldering a circuit board
  - not easy change/update functionality

Lots of Options

- Ardunio Family
  - Many to choose from (uno, mega, nano, ...)
    - Size
    - # of Digital I/O
    - # of Analog Inputs
  - Very inexpensive (~$2.00 - ~$20.00)
  - List of sensors is endless
  - Well supported (Open Source)
  - Development Environment is Free
  - Easy to get started

Lots of Options

- ESP8266
  - Very cheap (<$2.00)
  - Built in WiFi
  - Limited I/O (~8 Digital I/O, 1 Analog Input)
  - Can be used with Arduino
  - Supports a Flash File System (3MB)
  - Plug in for the Arduino IDE
  - A little more difficult to get started
  - Updates can be done over WiFi (plus/minus)

Lots of Options

- Raspberry Pi, BeagleBone
  - More processing power (similar to phone)
  - But larger and use more power
  - Run Linux from SD card
  - Improper shutdowns can corrupt file system
  - Much longer setup/learning curve
  - $15+
- Several others
  - If one of these doesn't suit your needs, a quick Google search may find one that does

Things to consider

- Micro-controllers are 5V or 3.3V
  - Inputs need to be conditioned
  - Outputs can drive ~40mA per pin
    - total 200mA
    - Typically need to use some form of buffering
- Possible EMI / RF interference from oscillator

Inputs/Sensors

- Voltage
- Temperature
- Magnetic Field
- Carbon Monoxide
- Vibration
- Acceleration
- Tilt

Inputs/Sensors

- Switches
- Capacitive Touch
- Range (ultasound, IR, LIDAR)
- RFId
- Ph
- Key pads
- Many, Many, Many more

Outputs

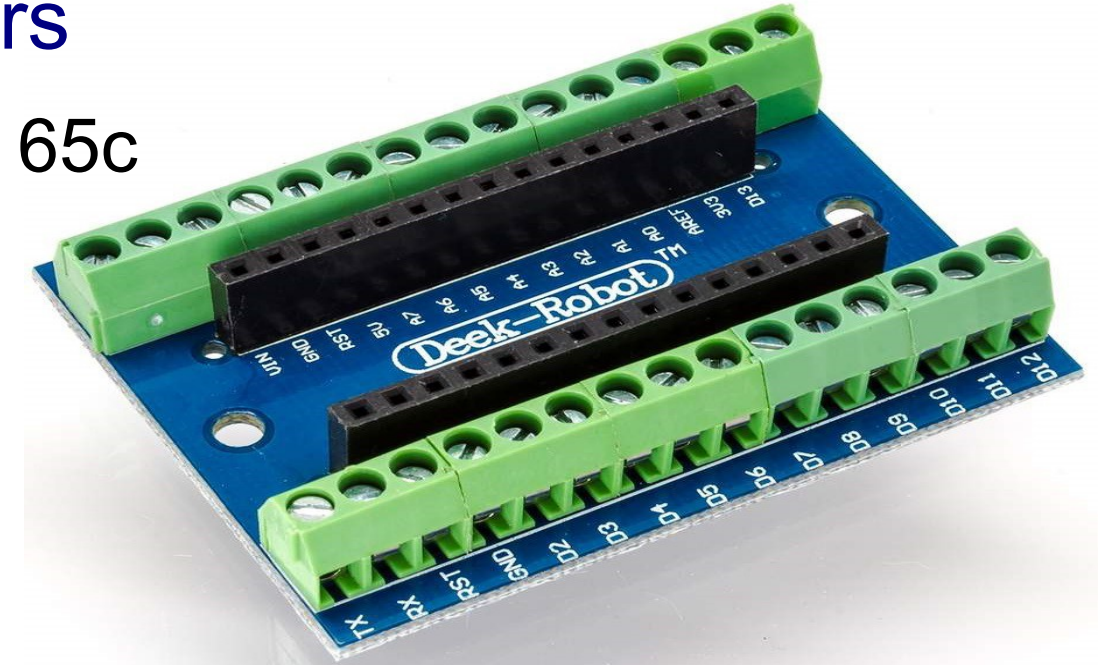
- Relay
- Audio (buzzers, beepers, voice)
- LEDs (single, multi-colour, strands)
- LCD Displays
- Servos
- Motor controllers (speed/stepper)

Connectivity

- USB
- RS-232 (TTL-level)
- Ethernet
- WiFi
- Bluetooth
- Some of these can be added by expansion boards or “shields”

How to Connect

- Solder to proto-board & point-point wire
- Custom Circuit Board
  - Inexpensive Options
- Terminal Adapters
  - As cheap as 65c



Programming

- Development Environments are Free
  - Well supported (forums)
  - Ask for donations when you download
- Programming languages
  - C/C++
  - LAU
  - Others

Programming

- Development Environments are Free
  - Well supported (forums)
  - Ask for donations when you download
- Programming languages
  - C/C++
  - LAU
  - Others

Best Practices

- Have a backup for any critical function
  - A wire coming loose from a circuit board could ruin your day!
- Consider all combinations of inputs
  - Can be useful to detect fault conditions
  - e.g. Gear up and down signals at same time
- KISS – Keep It Simple...
  - Switches perform function and provide input to the micro-controller vs micro-controller activating a relay

REMEMBER!

- Your system isn't FCC approved
  - Even if components are, wires can act as antennas for radiated/conducted noise
  - If it interferes with your radio, don't use it
- Its easy to make a mistake writing code
  - Verify every feature well!
  - Don't rely on it until it has been fully tested
  - Self diagnostics are highly recommended
    - Are inputs in proper state at power on?
    - Flash all “LEDs” so user can verify they work

How To Choose

- Match requirements to the device
  - # of inputs
  - # of outputs
  - Analog inputs
  - Processing requirements
  - Memory Requirements
  - Size

Some Ideas

- Engine Monitor
  - Arduino to monitor
    - Oil pressure
    - Temperatures
    - Battery voltage
    - Etc.
  - ESP8266
    - Stores data from Arduino on Flash File System
    - Download to PC periodically over WiFi

Some Ideas

- Airtime Logger
  - Use airspeed or accelerometer to detect take-off and landing time
  - Display time up on LCD
  - Could add prompts for maintenance, etc.

Some Ideas

- Landing evaluator
  - Airspeed and accelerometer
  - Measures how smooth you land
  - Audio Annunciator
    - Nice Landing!
    - Ouch

ALGMS

- Requirements
  - 8 Inputs
  - 10 Outputs
- Arduino Nano (<$2.00US)
  - 14 Digital I/O
  - 6 or 8 Analog Input (or Digital I/O)
  - 16 Mhz

ALGMS

- Arduino “Leonardo”
  - 20 Digital I/O
  - 12 Analog Inputs
  - 16 Mhz
  - Room for expansion

Some Ideas

- ADSB In
  - Raspberry Pi
  - USB SDR (Software Defined Radio) receiver
  - Google search for instructions

Questions?

- Where do you buy Arduinos and Sensors?
  - http://aliexpress.com - Chinese, 4-6weeks, wide selection
  - http://robotshop.ca - more expensive, 1 week
  - Sayal Cambridge – more expensive, in stock
  - Google to find best price/delivery options

Questions?

- Where do we get the Development software
  - http://arduino.cc - click on Download link
  - For the cheap arduinos, you may have to install additional drivers. Follow the instructions here:

    http://0xcf.com/2015/03/13/chinese-arduinos-with-ch340-ch341-serial-usb-chip-on-os-x-yosemite/

    with links to drivers for windows, linux and Mac.

Questions?

- How do I get started?
  - Buy a $2.00 Arduino nano, one or two simple sensors (eg. Ultrasonic range sensor), some Dupont breadboard cables, and possibly a breadboard. ~$10
  - Go to http://arduino.cc and click on LEARN ARDUINO at the left for instructions on installing the IDE and beginner guides
  - Look through the examples provided with the IDE
  - Google search for tutorials (many sensors have been documented on instructables.com)

Questions?

- How do you know what pins to connect the sensor to?
  - VCC (5V) & GND have specific pins on the Arduino
  - In most cases, you just need to make sure the pin # in software matches the pin number that the pin is connected to on the Arduino board.
  - There are some instances that specific pins are needed...look for an instructable/tutorial before buying a specific sensor if in doubt.