Progress Report #2

Anthony Llanos (Leader)
Jesús Luzón
Juan Lebrón
Jean C. Méndez

September 12, 2013

Contents

1 Block Diagram Version 2

A system block diagram was redesigned from scratch after considering the components that we would use for our implementation. After consulting professor Manuel Jimenez and director Raul Zapata, we determined components that will be used in the project and ended up with the following system block diagram:

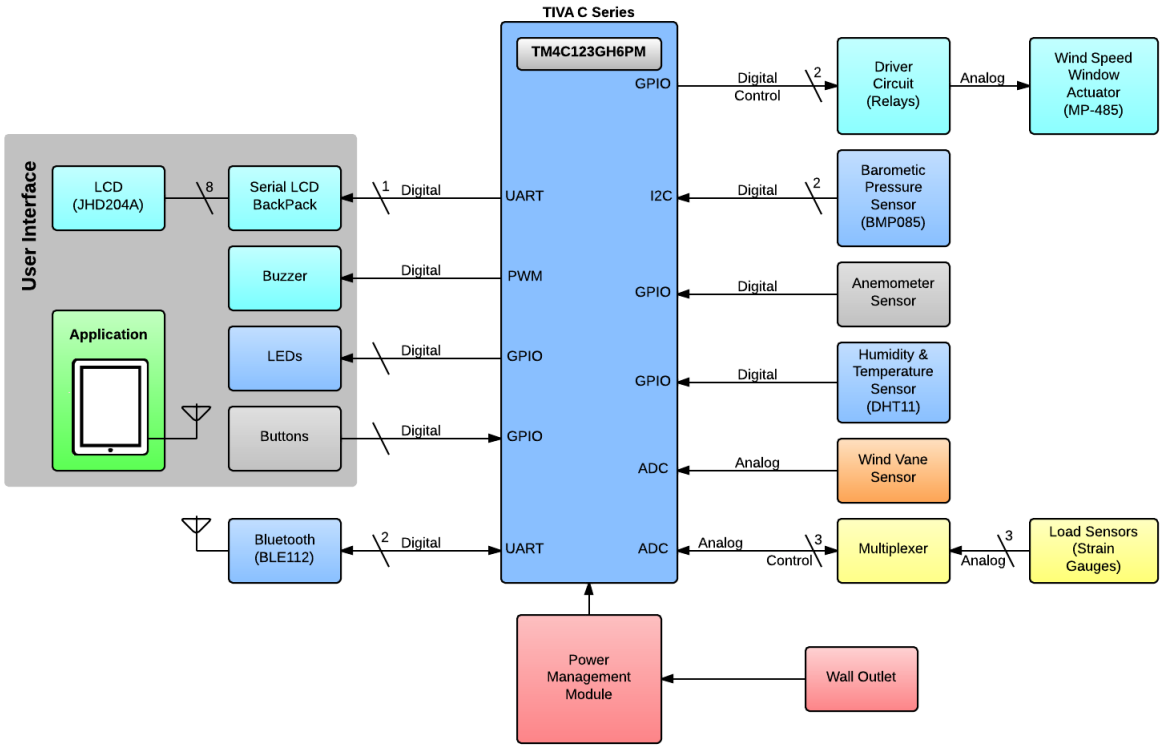

Figure 1: The second version of the AeroBal block diagram.

1.1 Material/Component search.

The following components were determined to be feasible for use in the project. The links to the webpages from which we will obtain or build these components are all listed at `aerobalmicro2.blogspot.com` as a post on September 11.

- Anemometer
  - To determine the speed of the wind inside the tunnel.
  - A computer fan may be used as an Anenometer. Due to initial resistance, the fan may have to be jump-started to eliminate initial resistance to move.
  - Interfacing can be achieved by using digital general purpose IO ports from the MCU.
- Barometic Pressure Sensor

  - Module Model BMP085 will be used.
  - Interfacing can be achieved using I2C protocol.

- Wind Direction Sensor
  - For the wind direction sensor (or wind vane), a potentiometer along with the tailvane may help in determining the direction as suggested by Manuel.
  - An ADC may be used for interfacing with this component.

- Strain Gauges
  - Found to be the ones used by weight scales for mail packages.
  - An ADC will be used for interfacing with this component.

- LCD
  - As an LCD we will use one based on the model JHD204A that uses a SPLC780D Controller.
  - UART protocol may be used with the Serial LCD Backpack Module that will convert the signal to de Parallel Bus that the LCD uses.

- Bluetooth Module
  - The model BLE112 from Bluegiga will be used.
  - Interfacing with this component can be achieved by using the UART protocol.

- Humidity and Temperature Sensor
  - A humidity sensor and a temperature sensor come as a single module in the package DHT11.
  - Interfacing can be achieved with general purpose IO ports.

- Buzzer, LEDs and Buttons.
  - Provided for a robust user interface in the hardware.
  - We can use common buzzers, as well as common LEDs and Buttons.

- Relays
  - The actuator that controls the windows for adjusting the wind speed (or the window actuator) is based on the model: MP-485. The control is based on a switch (Snap Acting or Floating Switch as in the specs).

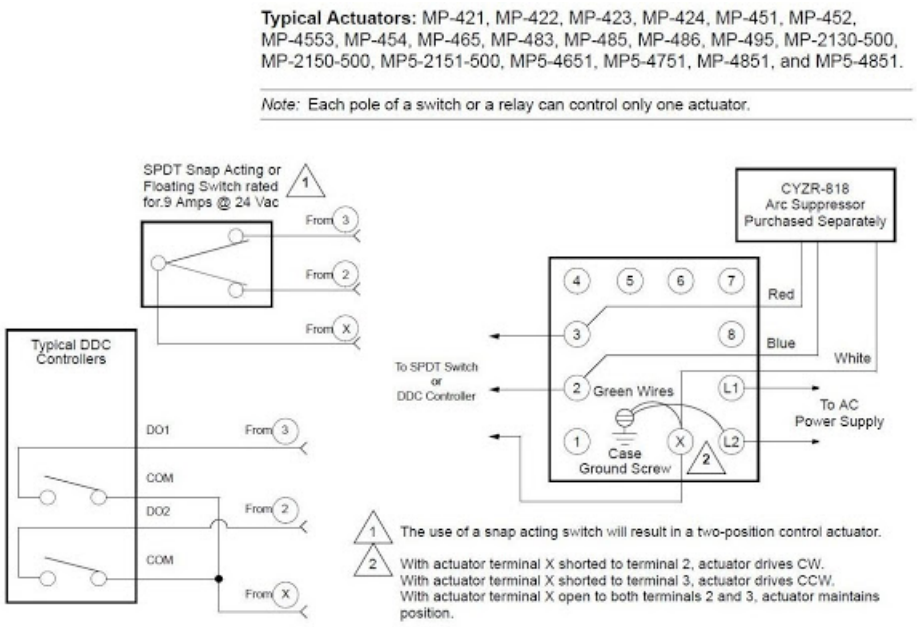

Figure 2: Actuator for controlling the window. This allows for control of the speed of the wind.

– Relays can be used to achieve the logic of this tri-state switch in the following manner, and this approach will be used if approved:

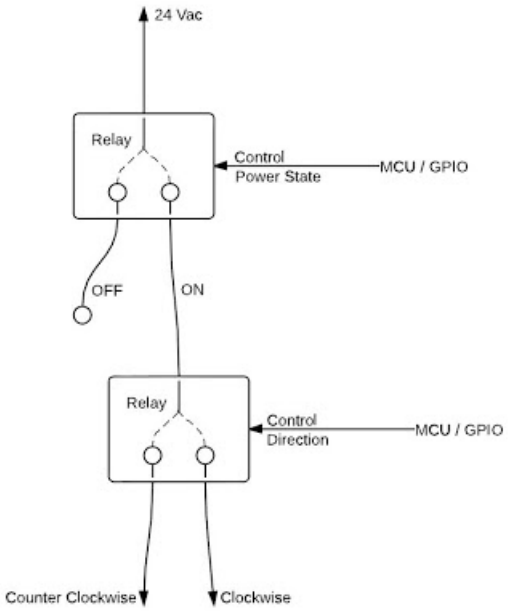

Figure 3: A possible implementation of the circuit to enable control of the window actuator

2 MCU Selection

For the selection of our microcontroller, we started by considering various factors. These were:

- **Architecture** - At first we did not consider the architecture was a factor we should heavily worry about. However, once we realized the labs of the course will be made exclusively with assembly language (ASM), we had to verify how said language varied between architectures. The ASM of the ARM architecture is different from that of the MSP430 architecture, so we had to learn various things of the language if we wanted to select an MCU with ARM architecture.

- **Data Bit Length** - *Aerobal* will be doing constant calculations and transmission of data. In theory, the data the system will process will not be excessively large. Therefore, the group thought that 16 bits should be enough. *Warning*: We will use a prototype miniature wind tunnel in the lab. *Aerobal* will work with out model, which will have a small home fan. It should work just as well with the wind tunnel in the Wind Tunnel Laboratory. However, we are aware that the data in the real wind tunnel will be bigger than the one we get in the lab. For this reason, more than 16 bits might be good just to make sure we have enough space.

- **Universal Asynchronous Receiver Transmitter (UART) Pins** - *Aerobal* needs two UART pins. One for Bluetooth connectivity, and another one for the LCD screen.

- I^2C **Pins** - We need an MCU with I^2C pins to interface with the barometric pressure sensor. Two pins should suffice.

- **Clock Speed** - Our slowest component will be the temperature sensor, which we have seen usually run at around $10\mu s$. Thus, we need an MCU that has a clock speed of at least $1/10\mu s = 100kHz$.

- **Analog-to-digital Converters (ADC)** - The wind vane and load sensors both are analog and need analog-to-digital converters to work with our MCU. Ideally, we would need four converters because there is one vane sensor and three load sensors. If not, then we would have to multiplex one of the converters.

- **General Purpose Input/Output Pins (GPIO)** - The following components of our embedded systems are connected through GPIO:

  1. Wind Speed Actuator
  2. Anemometer Sensor
  3. Humidity Sensor
  4. LED's of User Interface
  5. Buttons of User Interface

  Thus, we will probably need more than 6 GPIO's for *Aerobal*.

- **Cost** - Besides the cost of the MCU, we also need to consider if we wanted to have a JTAG debugger. This was a huge factor, because not many MCU's come in a board (or a "launchpad") with an integrated JTAG debugger. Thus, if it does not come with it, we would need to buy a development board to have this functionality.
- **FLASH and RAM** - Although having big amounts of memory in our system is always great to have, it is not something we necessarily need. The MCU we select should be able to do all of it's processing with 8 kB of RAM. As of right now, we do not have any plans of storing data in FLASH memory.

	Tiva	MSP430F5528	PIC	Piccolo TMS320C2000
Architecture	ARM	MSP430	PIC	C28x
Data Bit Length	32	16	16	32
UART Pins	8	2	2	1
I^2C Pins	4	2	2	1
Clock Speed	80 MHz	25 MHz	7.37 MHz	100MHz
ADC's	2	1	2	1
GPIO	43	47	85	35
Cost (w/ JTAG)	$13	$175	$155	$100
RAM	32kB	8kB	16kB	12kB
FLASH	256kB	128kB	256kB	64kB

After considering all the points mentioned above, it is clear that our best option would be the Tiva C Series MCU. Primarily, it come with a JTAG debugger in it's launchpad (unlike the MSP 430), so we don't need to spend $149 on it's development board. Besides that, it also has more bits, more pins, higher clock speed, and more converters than most of the other options.

We add to that it's price of $13, so we can buy various MCU's in case we have accidentally burn the ones we have. Our main problem with this model is it's ARM architecture, so programming it in ASM will not be as easy as it would have been with the MSP 430.

MCU Selected: Tiva C Series TM4C123GE6PM

3 Operating Chart

For the operating chart, we determined the sequence of steps that the system goes through to perform its functionality. In the end we obtained the following procedures:

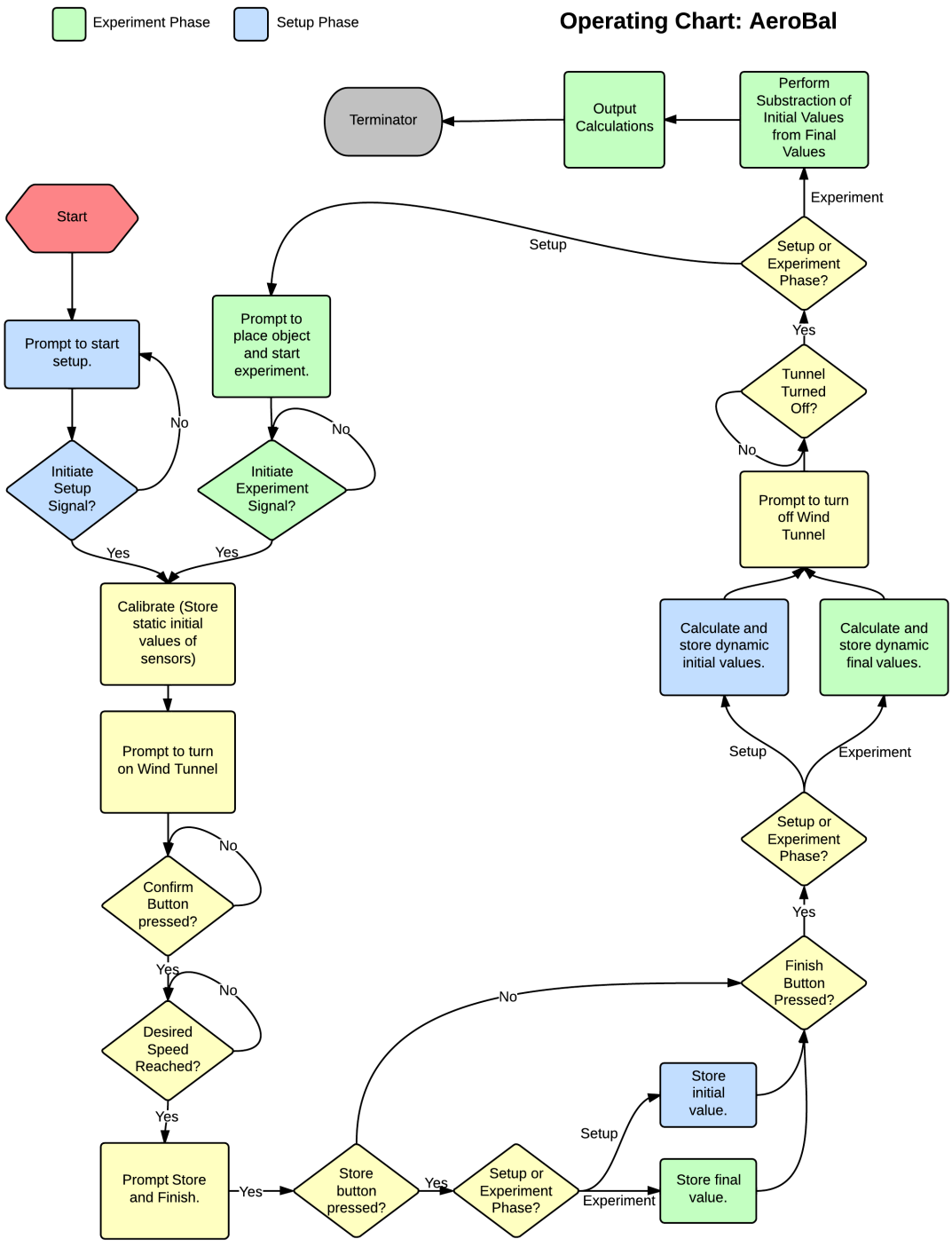

Figure 4: The set of procedures AeroBal goes through.

The set of procedures can be *summarized* as setup phase and an *experiment* phase. For both phases the system must perform the same procedure, and after performing both procedures use the data obtained to compute the desired result. This is because the procedure must be executed for the base of the object and then the base along the object to later perform substraction of the results and obtain the values belonging to the object by itself.

The procedure is composed of the following steps:

1. Prompt to place the component (either the base for the setup phase or the object in the base for the experiment phase)
2. Wait for the confirmation to start the phase.
3. If confirmed, start the phase by obtaining the *reference* values of the strain sensors.
4. Prompt and await for wind tunnel to turn on.
5. Once confirmed that the wind tunnel is turned on, await either store value or finish button to be pressed.
    - If in setup phase, the value will be stored as an initial value.
    - If in experiment phase, the value will be stored as a final value.
6. If finish button is pressed the average value of the stored values will be stored as the initial or final value.
7. If in the setup phase, the system will prompt to place the object back at step 1. If not then it continues to the next step.
8. The system computes the final values from both by substracting the initial values from the final values.
9. The system outputs the results and finishes.

This procedure is followed on the hardware user interface as well as on the software user interface with the exception that on the software user interface the values are stored on the device/tablet and can be viewed later by the user.

The system has a configuration interface both in the hardware and software interfaces that allows for the configuration of the speed in the wind tunnel.