Program Design

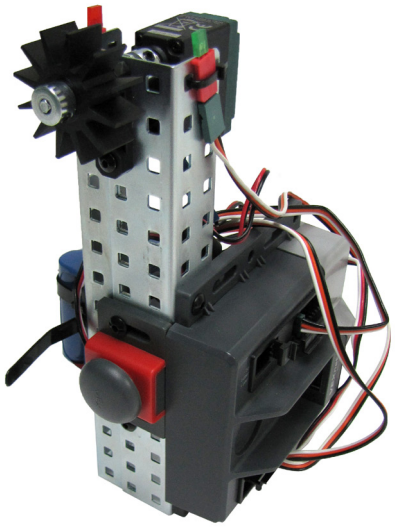
Planning your program may occur after you have sketched or built your physical device. It may also occur before or at the same time depending on the your challenge. Regardless it is good practice to do some planning for your program before writing code. This document outlines a strategy for a simple example to show the process. Please review the reference **Behaviors**, to familiarize yourself with basic, simple, and complex behaviors.

The steps below should be documented in your engineering notebook . Some of the information will then be transferred to the PLTW ROBOTC program template.

1. Describe the task or overall goal that your program will accomplish. This may be described as one or more **complex behaviors**.

Example: A fan will run until someone needs it to stop. There will be a warning light as a safety device before the fan turns on and another light to indicate that the fan has stopped.

Note: This text will be used for the **Task Description** in the PLTW ROBOTC program template



Creating Pseudocode

As you begin to break down your behaviors into individual actions do not worry about syntax or which commands will be used with ROBOTC. Simply describe them in short phrases such *turn a motor on for three seconds* or *follow a line until running into a wall*.

2. For each complex behavior break it down into **Simple Behaviors** line by line in the order that each should happen. Try to describe actions and what prompts each action to continue, start, stop, etc.

Example:

A warning light comes on before the fan starts for three seconds

The fan turns on and runs until a button is pressed

A different light turns for three seconds before the program stops

3. For each simple behavior, break it down further to **Basic Behaviors**. Try to think in terms of what each input and output component will be on your device.

Example:

Program begins

Light 1 (LED 1) turns on

for three seconds

Fan (Motor 1) turns on

Until a button (bumper switch)is pressed

Light 2 (LED 2) turns on

for 3 seconds

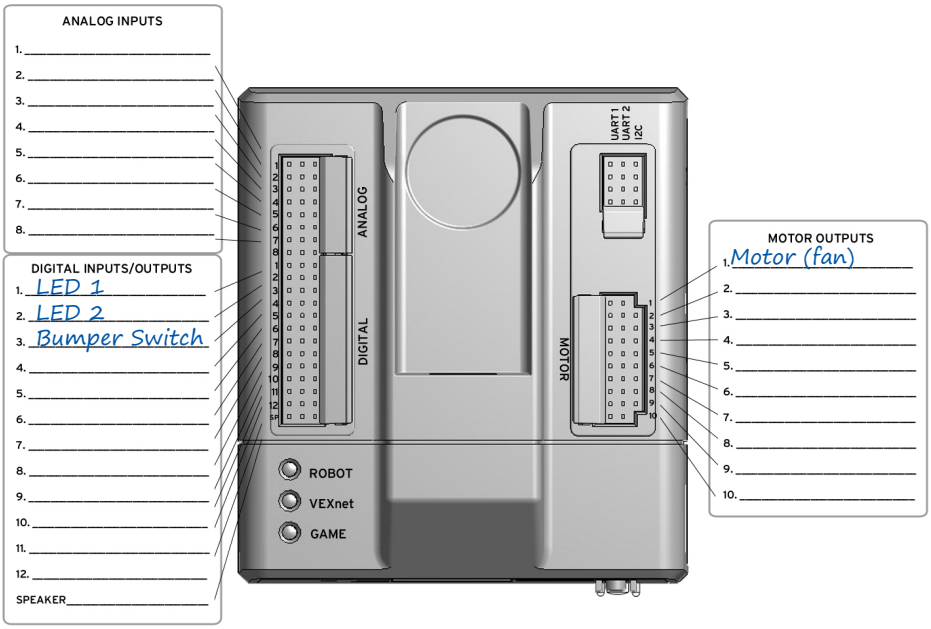
Program ends

Note: This text will be used for the **Pseudocode** in the PLTW ROBOTC program template

Program Design Identify Inputs and Outputs

4. Now that you know what inputs and ouputs you will need, identify which ports each will be plugged into on the Cortex. Pay attention which sensors are analog and which are digital. Below is a sketch of a possible configuration for the example on the previous page.

Example:



Note: The last page of this document contains a clean image of the Cortex that you can label, then cutout and attach in your engineering notebook for your own projects.

Program Design PLTW ROBOTC Program Template

Note: Be sure the Cortex you are using has been updated with the Master CPU and ROBOTC Firmware. Refer to the reference **Firmware Over USB** to acquire detailed instruction for this procedure.

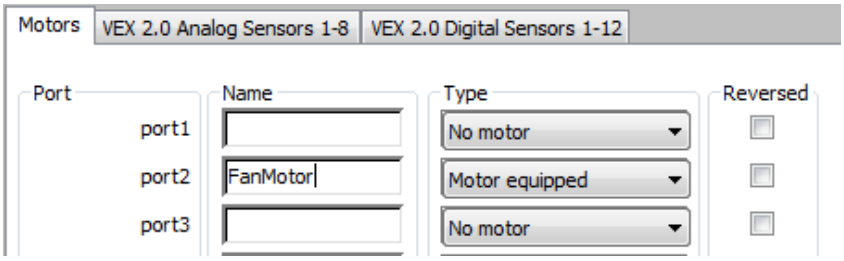
5. Open ROBOTC and open the Sample Program PLTWTemplate.
6. Use your initial description (Complex Behaviors)of your overall goal for the program for the Task Description.
7. Copy your final pseudocode (Basic Behaviors) for the Pseudocode section of the PLTW ROBOTC program template.
8. It is recommended that you include your pseudocode mostly in tact as comments beside programming commands.

Example:

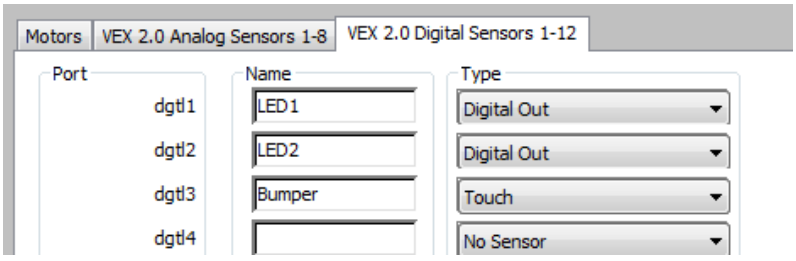
```
/*
  Project Title:
  Team Members:
  Date:
  Section:


  Task Description:

A fan will run until someone needs it to stop. There will be a warning light
as a safety device before the fan turns on and another light to indicate that the
fan has stopped.

  Pseudocode:

Program begins
Light 1 (LED 1) turns on
for three seconds
Fan (Motor 1) turns on
Until a button (bumper switch)is pressed
Light 2 (LED 2) turns on
for 3 seconds
Program ends

*/

task main()
{
 //Program begins
 //Light 1 (LED 1) turns on
 //for three seconds
 //Fan (Motor 1) turns on
 //Until a button (bumper switch)is pressed
 //Light 2 (LED 2) turns on
 //for 3 seconds
 //Program ends
}
```

Program Design PLTW ROBOTC Program Template Cont.

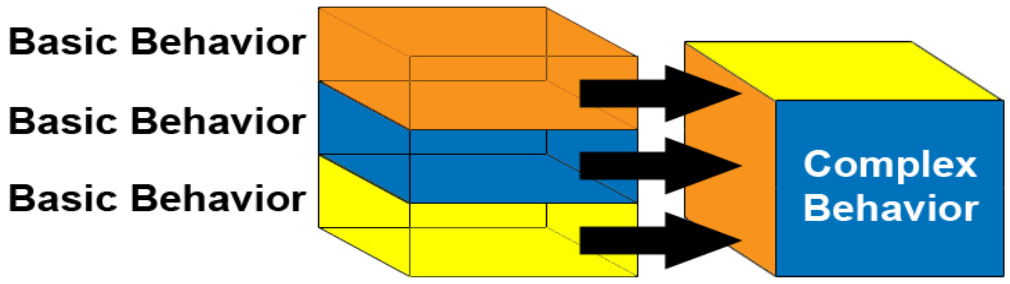
9. Identify all inputs and outputs in the Motors and Sensors Setup window.



10. Use the Debugger to confirm that all inputs and outputs are working as expected. Refer to the reference **Debugger** to learn more about these functions.



Program Design PLTW ROBOTC Program Template Cont.



Remember many basic behaviors generally come together to create a complex behavior . You can solve simple and basic behaviors one at a time, and troubleshoot them as they come together to form a complex behavior.

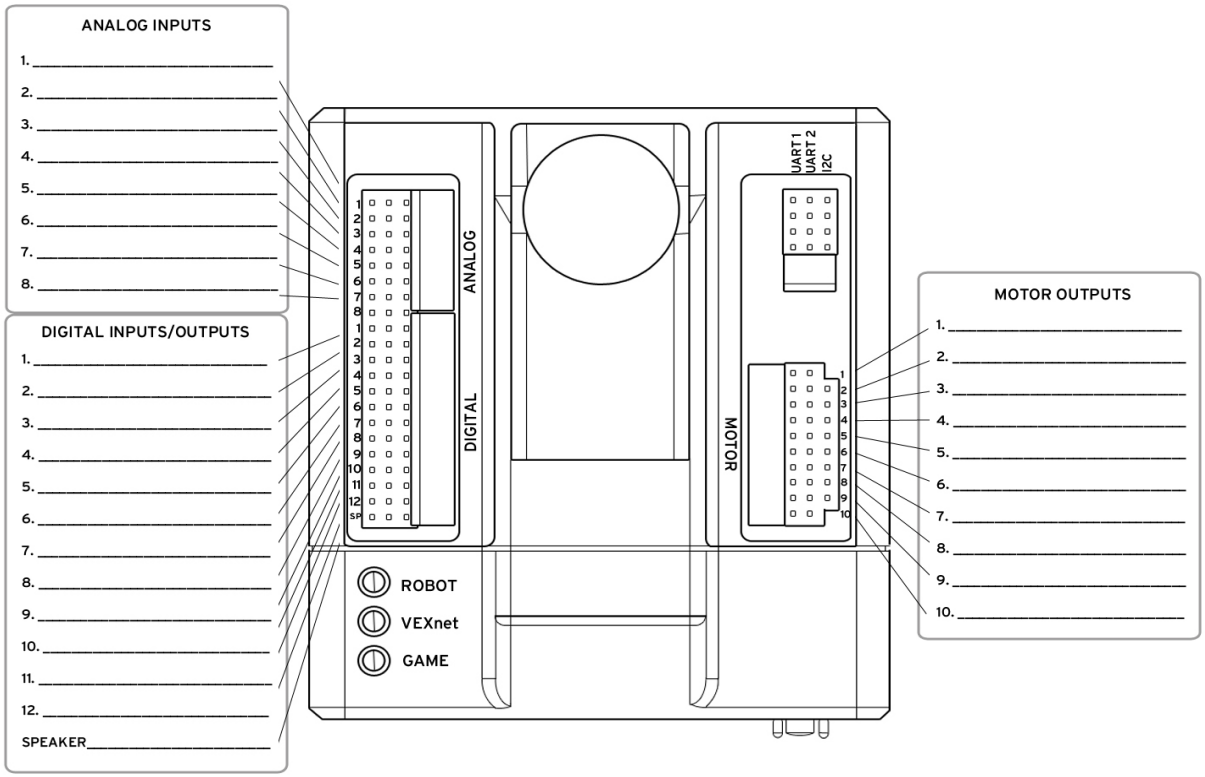
Test and debug the combined program. Make sure your behavior functions as intended within the program. Many times, you will need to make adjustments to compensate for orientation, momentum, or other unforeseen factors as they begin to work together

11. Code and test small behaviors or sets of behaviors individually and edit / add comments as you build your code.

```
33  task main()
34  {
35                                  //Program begins
36  turnLEDOn(LED1);                //LED1 turns on
37  wait(3);                        //for three seconds
38  turnLEDOff(LED1);               //LED1 turns off
39
40  //FanMotor turns on
41  //Until Bumper is pressed
```

12. Continue programming and testing one behavior at a time. To test individual behaviors as you go temporarily turn sections of code into comments using /* followed by */.

```
33  task main()
34  {
35                                  //Program begins
36  /*
37  turnLEDOn(LED1);                //LED1 turns on
38  wait(3);                        //for three seconds
39  turnLEDOff(LED1);               //LED1 turns off
40  */
41
42  startMotor(FanMotor, 127);      //FanMotor turns on
43  untilBump(Bumper);              //Until Bumper is pressed
44  stopMotor(FanMotor);            //FanMotor turns off
45
46  //Light 2 (LED 2) turns on
```


ANALOG INPUTS
1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
DIGITAL INPUTS/OUTPUTS
1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
SPEAKER_____
ANALOG
DIGITAL
MOTOR
UART 1
UART 2
I2C
ROBOT
VEXnet
GAME
MOTOR OUTPUTS
1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____