This assignment includes (3) problems.

|  |  |
| --- | --- |
| **Purpose of this assignment** | This assignment introduces you to the TI kits and the Energia IDE. You will be introduced to basic Aurdino code to execute commands inside of Energia using various built-in features of the TI kits.  TI Kits will be utilized throughout this course to solve various problems while building your skills to allow teams to utilize them as part of their solution to their design project. |
|  |  |

**Relevant Course Resources**:

|  |  |
| --- | --- |
| Pre-Class Videos | * None yet |
| Course Resources | * Getting Started with Energia and the TI Kits * Block Diagram Basics |
| Lecture Slides | * Class ## Slides |

|  |
| --- |
| **Communication Errors between the TI Kits and Your Computer:**   1. Make sure that your TI Kit is plugged in correctly using the provided Micro USB to USB A cable. There should be a green power LED lit up when this occurs. 2. The COM port selected in Energia is not the correct port. While it is generally the highest port number available. This is not always the case. All COM Ports should be tried if there are communication issues. 3. Ensure all drivers have been installed. For instructions on how to install, please see Step 2. 4. Restart Energia IDE 5. Restart your computer (Should not be required, but can help depending on your computer’s settings)   **Submission Instructions:**   1. Re-name your answer sheet as, **ENGR131\_A03\_*yourlogin*.docx**, where *yourlogin* is your *Purdue Career Account* login. 2. Save your files to your **Purdue Career Account** (This is your Purdue storage space. For more information see <https://www.itap.purdue.edu/connections/careeraccount>) 3. Submit your work through the designated **Brightspace Assignment Drop box at** [https:/purdue.brightspace.com/](https://mycourses.purdue.edu/) |

|  |  |
| --- | --- |
| **Problem 1** | **Uploading a Sketch to the TI Kit Board** |
| **Goal** | This problem tests your ability to upload a sketch to the TI Kit board demonstrating that the TI kit can communicate with the computer. To do this, you will need to complete the following:   1. Open the Example/Basics/Blink file in Energia. Include Screenshot of location 2. Compile and upload the Blink sketch to your TI Kit.    1. **HINT:** If this works, two lights should be illuminated on your TI Kit.   Then answer the following questions:   1. Draw (by hand or via a computer) a block diagram of your set up on your answer document. 2. Take a picture of your TI Kit with the blinking red light lit up and submit this in the answer document. |
| **Solution: Block Diagram** |  |
| **Solution: Picture** |  |
| **Reference: CODE** | The Blink code should be built in, but in case it isn’t, please use the following code in a blank sketch:  /\*  Blink  The basic Energia example.  Turns on an LED on for one second, then off for one second, repeatedly.  Change the LED define to blink other LEDs.    Hardware Required:  \* LaunchPad with an LED    This example code is in the public domain.  \*/  // most launchpads have a red LED  #define LED RED\_LED  //see pins\_energia.h for more LED definitions  //#define LED GREEN\_LED    // the setup routine runs once when you press reset:  void setup() {  // initialize the digital pin as an output.  pinMode(LED, OUTPUT);  }  // the loop routine runs over and over again forever:  void loop() {  digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)  delay(1000); // wait for a second  digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW  delay(1000); // wait for a second  } |

|  |  |
| --- | --- |
| **Problem 2** | **Communication between the TI Kit and Your Computer: Serial Port** |
| **Goal** | This problem tests your ability to upload a sketch to the TI Kit board demonstrating that the TI kit can communicate with the computer. To do this, you will need to complete the following:   1. Download the Serial\_Data\_Test.ino file from [Github](https://github.com/paulleywalnuts/ENGR131-TIKITS/tree/9ca7d7ad5660cab2a773decf209c0e6109a867eb) and open it in the Energia IDE. 2. Compile and upload the Serial\_test sketch to your TI Kit. 3. Open the Serial Monitor located in the Tools Menu to complete question 2. 4. Close the Serial Monitor and open the Serial Plotter located in the Tools Menu to complete question 3.   Then answer the following questions:   1. Draw (by hand or via a computer) a block diagram of your set up on your answer document. 2. Take and upload a screenshot of the Serial Monitor window when 25 data points have been taken. 3. Take and upload a screenshot of the Serial Plotter window when 500 seconds of data have been taken. |
| **Solution: Block Diagram** |  |
| **Solution: Serial Monitor** |  |
| **Solution: Serial Plotter** |  |
| **Reference: CODE** | #define sensorPin A2  void setup() {  // initialize serial communication at 9600 bits per second:  Serial.begin(9600);  //analogReference(INTERNAL);    }  void loop() {  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):  // get the temperature and convert it to celsius  float reading = analogRead(sensorPin);  float voltage = reading \* 5.0 / 1024.0;  float temp = voltage \* 100 ;    // print out the value you read:  Serial.print(temp);    Serial.print(" \xC2\xB0");  // print out the value you read, and skip next line  Serial.println("C");  delay(1000);  } |

|  |  |
| --- | --- |
| **Problem 3** | **Basic Circuit with the TI Kit** |
| **Goal** | Now that you understand the basics of the TI Kit, it is time to begin using sensors to solve various problems. In this first problem, you will simply be building a circuit using the TI Kit to play the Purdue Fight Song. To do this, you will need to complete the following:   1. Download the buzzer\_purdue\_fight\_song.ino file from [Github](https://github.com/paulleywalnuts/ENGR131-TIKITS/tree/9ca7d7ad5660cab2a773decf209c0e6109a867eb) and open in Energia. 2. Before compiling and uploading the sketch, first you need to add to the board.    1. Connect the Grove Starter Kit Buzzer to J14 of the Boosterpack using the four-prong connector cable. INCLUDE PICTURE OF Assembly    2. Connect the Boosterpack underneath the TI Kit board. INCLUDE PICTURE OF Assembly 3. Upload and compile the Buzzer\_Fight\_Song file.   Then answer the following questions:   1. Draw (by hand or via a computer) a block diagram of your set up on your answer document. 2. Record a sound clip of your buzzer playing the Purdue Fight Song and upload it along with your answer document. |
| **Solution: Block Diagram** |  |
| **Solution: Sound Clip** | Please submit the audio clip along with the answer document when submitting this assignment. |
| **Reference: CODE** | /\*  Grove Buzzer  The example uses a buzzer to play melodies. It sends a square wave of the  appropriate frequency to the buzzer, generating the corresponding tone.    The circuit:  \* Buzzer attached to Pin 39 (J14 plug on Grove Base BoosterPack)  \* Note:    This example code is in the public domain.    http://www.seeedstudio.com/depot/Grove-Buzzer-p-768.html    \*/    /\* Macro Define \*/  #define BUZZER\_PIN 39 /\* sig pin of the Grove Buzzer \*/  int length = 59; /\* the number of notes \*/  char notes[] = "dewgabbcccgatb bbagabbaewgwea ddewgabbbccgab ewgedgbdebagg "; /\*notes in the song. Use a space for rests\*/  int beats[] = { 2, 2, 2, 3, 1, 2, 2, 2, 1, 1, 2, 1, 1, 5, 1, 4, 2, 2, 3, 1, 2, 2, 2, 1, 1, 2, 1, 1, 5, 1, 3, 1, 2, 2, 3, 1, 2, 1, 1, 2, 2, 2, 2, 5, 1, 3, 1, 2, 2, 2, 2, 2, 2, 3, 1, 3, 1, 5, 1 }; /\*length of each note. 1 = quarter note\*/  int tempo = 200;  /\* the setup() method runs once, when the sketch starts \*/  void setup()  {  /\* set buzzer pin as output \*/  pinMode(BUZZER\_PIN, OUTPUT);  }  void loop()  {  //Loop through each note  for(int i = 0; i < length; i++)  {  //space indicates a pause  if(notes[i] == ' ')  {  delay(beats[i] \* tempo);  }  else  {  playNote(notes[i], beats[i] \* tempo);  }  delay(tempo / 2); /\* delay between notes \*/  }  }  /\* play tone \*/  void playTone(int tone, int duration)  {  for (long i = 0; i < duration \* 1000L; i += tone \* 2)  {  digitalWrite(BUZZER\_PIN, HIGH);  delayMicroseconds(tone);  digitalWrite(BUZZER\_PIN, LOW);  delayMicroseconds(tone);  }  }  /\* List of the notes in the song \*/  /\* w = F sharp, t = A sharp \*/  char names[] = { 'c', 'd', 'e', 'f', 'w', 'g', 'a', 't', 'b', 'C' };  /\* Match the notes to the wavelength of the soundwave in cm \*/  /\* Note: This code assumes that sharps are half steps between notes \*/  int tones[] = { 1915, 1700, 1519, 1432, 1354, 1275, 1136, 1075, 1014, 956 };  void playNote(char note, int duration)  {      // play the tone corresponding to the note name  for (int i = 0; i < 10; i++)  {  if (names[i] == note)  {  playTone(tones[i], duration);  }  }  } |