This in-class activity includes (3) problems.

|  |  |
| --- | --- |
| **Purpose of this in-class activity** | This activity introduces you to the TI kits and the Energia IDE. You will be introduced to basic Arduino 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 |
| Course Resources | * Getting Started with Energia and the TI Kits * Block Diagram Basics |
| Lecture Slides | * Class 02A 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 in [Getting Started](https://engineering.purdue.edu/fye_i2i/ti-kit-activities-getting-started/). 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\_ICA02A\_*Team##*.docx**, where *team##* is your assigned ENGR 131 team. 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 In-Class Activity Drop Box at** [https:/purdue.brightspace.com/](https://mycourses.purdue.edu/) |

|  |  |
| --- | --- |
| **Task 1** | **Communication between the TI Kit and Your Computer: Serial Port** |
| **Goal** | This task 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 Sketchbook/ENGR 131/Activity 1 – Introduction/Task1b\_Serial\_Data\_Test file in Energia. 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.   This is a picture that demonstrates where to find the Serial Monitor functionality. It is located in the tools menu of the Energia IDEA and is the fifth item down the menu.  Figure 2: Serial Monitor Location   1. 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);  } |

|  |  |
| --- | --- |
| **Task 1c** | **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 task, you will simply be building a circuit using the TI Kit to play a song that you should know. Can you guess it? To do this, you will need to complete the following:   1. Open the Sketchbook/ENGR 131/Activity 1 – Introduction/Task1c\_Song file 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.   This picture shows the Grove Starter Kit Boosterpack with the Grove Starter Kit buzzer connected to it.  Figure 3: Grove Starter Kit Buzzer Boosterpack Connection   * 1. Connect the Boosterpack underneath the TI Kit board.   Figure 4: Top View of Boosterpack to TI Kit Connection    Figure 5: Side View of Boosterpack to TI Kit Connection   1. Upload and compile the Task1c\_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 activity. |
| **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);  }  }  } |