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 01B 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** | **Uploading a Sketch to the TI Kit Board** |
| **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/Task1a\_blink file in Energia.   This is a screenshot of the Energia IDE software showing how to access the Task1a_Blink file through the sketchbook filepath system.  Figure 1: TI Kit Basics Filepath Example through Sketchbook   * 1. If you don’t see the ENGR 131 folder in your sketchbook options, please see the [Getting Started page](https://engineering.purdue.edu/fye_i2i/ti-kit-activities-getting-started/) for download and instructions.  1. 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  } |