# FRDM-KEA LAB EXAMPLES

## SUPPORTS FRDM-KEAZ128 AND FRDM-KEAZ64

ULTRA-RELIABLE MICROCONTROLLERS (MCU) FOR INDUSTRIAL AND AUTOMOTIVE
FEB, 2020

**NXP**

SECURE CONNECTIONS
FOR A SMARTER WORLD

# Index

- Board Overview
- Lab Examples
  1. Blink LEDs using Loops : Blinky Loop
  2. Blink LEDs using FlexTimer Module : Blinky FTM
  3. Blink LED using Real Time Clock : Blinky RTC
  4. Getting Familiar with Pulse Width Modulation : Dimmer PWM
  5. Getting Familiar with Analog to Digital Converter : ADC
  6. Combine FTM and ADC : Blinky FTM ADC
  7. Getting Started with Key board Interrupt : KBI
  8. Communicate using Universal Asynchronous Receiver Transmitter (UART)
  9. Communicate using Bluetooth
  10. Communicate using Local Interconnect Network (LIN)
  11. Control LCD Arduino Shield with KEA

# BOARD OVERVIEW

# Board Overview
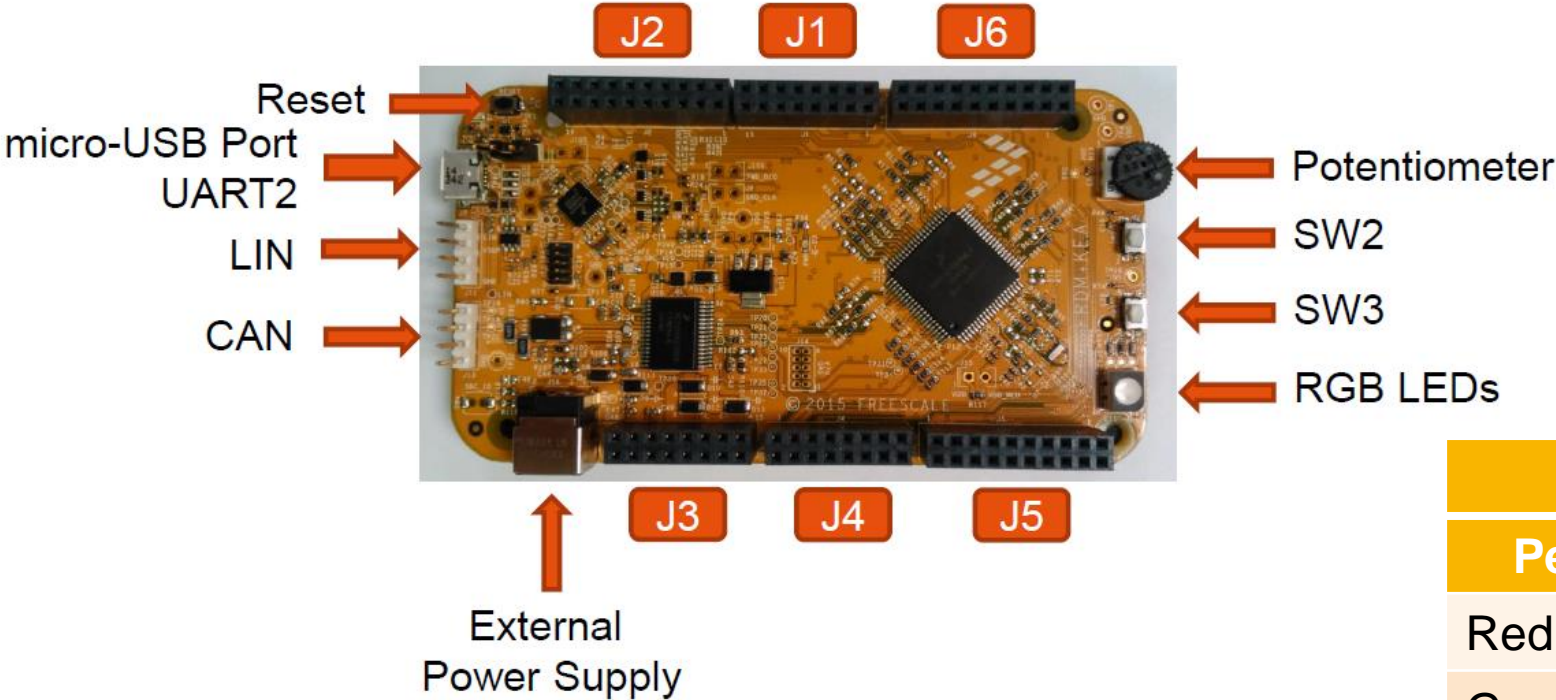


- *Note : for more information please see Schematics*

| Pin MappingTable | |
|---|---|
| **Peripheral Name** | **Pin Number** |
| Red LED | PTH0 |
| GreenLED | PTH1 |
| Blue LED | PTE7 |
| SW2 | PTE4 |
| SW3 | PTE5 |

# LAB EXAMPLES

# Lab1. Blink LEDs using Loops : Blinky Loop

## Things to Learn:

- Configure GPIO

- Initialize Clock in FEI mode



RGB LEDs

## Overview:

- This Lab demonstrates how to define the GPIO port pins and how to manipulate them to change their direction and there value(high/low).

- More over, it explains basic configuration to set clock into FEI –FLL Engaged Internal mode to generate the core frequency of 40 MHz and Bus Frequency of 20MHz.

## Hardware Connections:

- No Hardware connections required

## Observation:

- Running this example, changes the color of RGB LEDs periodically. It uses simple loop to generate the delay and to repeat the pattern.

# Lab2. Blink LEDs using FlexTimer Module : Blinky FTM

## Things to Learn:

- Make GPIO configuration generic
- Initialize FTM in Output Compare mode
- Use Interrupts



RGB LEDs

**Overview:**

- This Lab demonstrates how to make GPIO functions generic by making header file. For example, make functions to configure any GPIO pin input, output, set, reset etc.
- Further, it teaches how to use FTM module in Output Compare mode and how to utilize periodic interrupt to generate patterns on LEDs.

**Hardware Connections:**

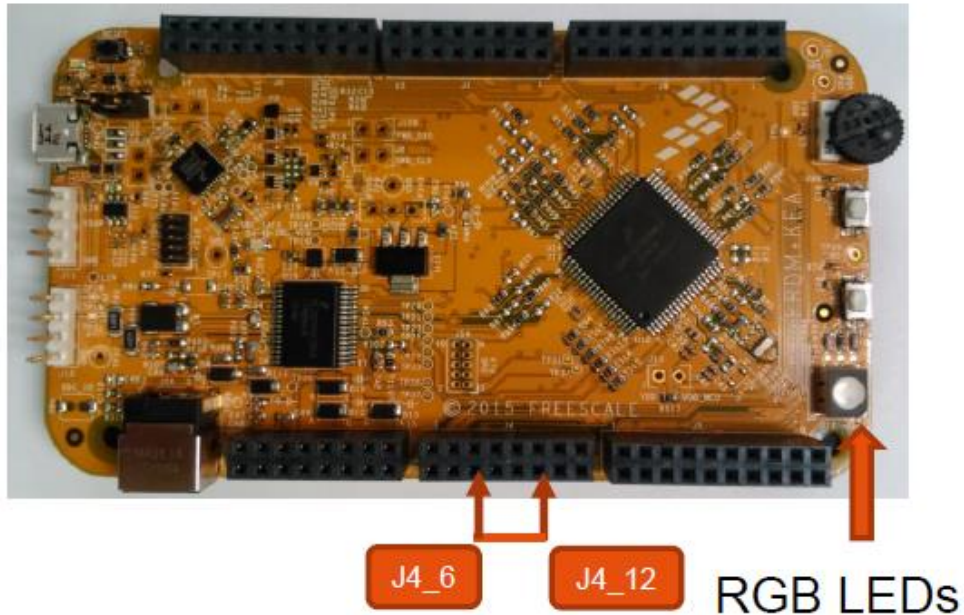- No Hardware connections required

**Observation:**

- Running this example, changes the color of RGB LEDs . It uses interrupt to jump to the subroutine and perform the change in LED color when counter value match occurs. So, blinking of different LEDs can be observed.

# Lab3. Blink LEDs using Real Time Clock : Blinky RTC

## Things to Learn:

- Initialize RTC



**Overview**:

- The RTC can be used for time-of-day, calendar or any task scheduling functions. It can also serve as a cyclic wake-up from low-power modes.

- This example has configured to generate precise waveform on external pin to show that it can be used as a clock source for some other devices.

## Hardware Connections:

- Connect PTC5(J4_6) to PTE7(J4_12)

## Observation:

- Running this example will toggle the Blue LED periodically.

# Lab4. Getting Familiar with Pulse Width Modulation : Dimmer PWM

## Things to Learn:

- Initialize FTM in Pulse Width Modulation mode



RGB LEDs

**Overview:**

- PWM feature can be used in BLDC motor control applications, DC motor control, lighting control etc applications.

- Example teaches how to use FTM module in Edge-aligned PWM mode and how to utilize interrupt to realize different functionality through call to subroutine..

## Hardware Connections:

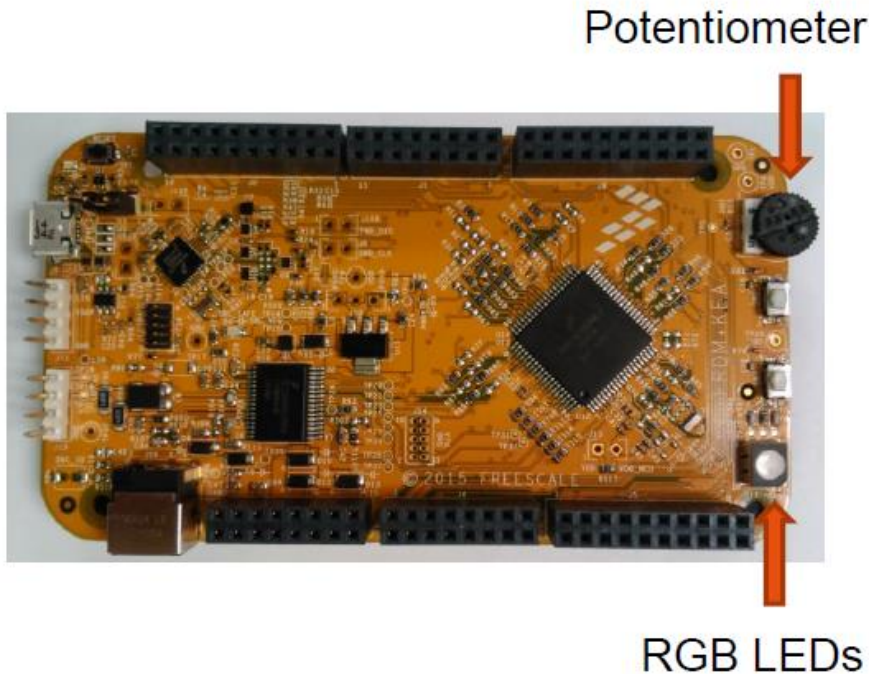- No Hardware connections required

## Observation:

- Running this example, changes the color of RGB LEDs periodically. It uses simple loop to generate the delay and to repeat the pattern.

# Lab5. Getting Familiar with Analog to Digital Converter : ADC

## Things to Learn:

- Initialize & Read ADC Module



Potentiometer

RGB LEDs

## Overview:

- ADC is used to measure some real world values like Temperature, Pressure, Resistance etc.
- Example teaches how to use ADC to read the port value and generate a feedback to indicate the measured reading.

## Hardware Connections:
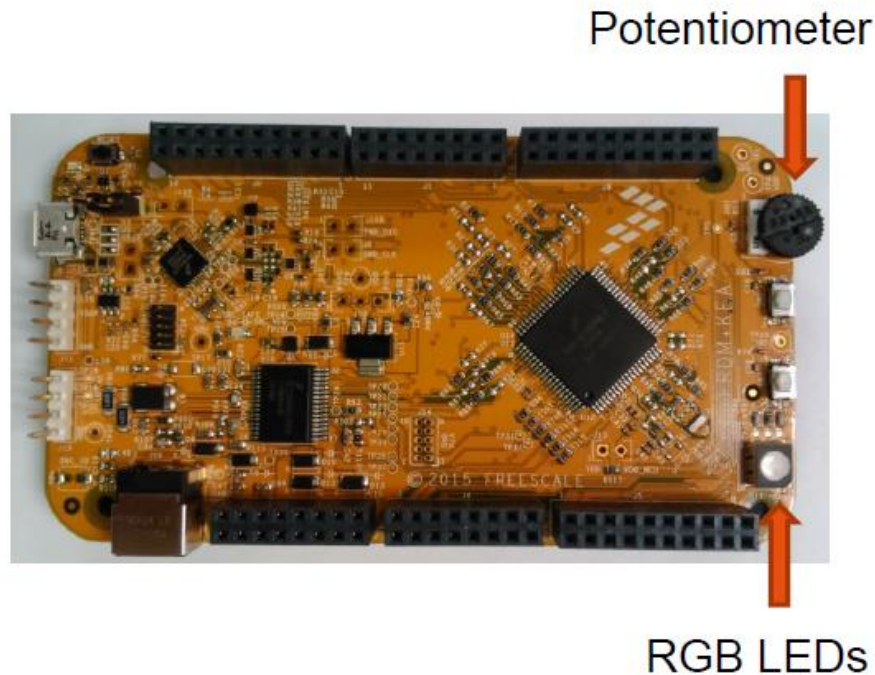
- No Hardware connections required

## Observation:

- Potentiometer is connected to the ADC CH10. Whenever potentiometer crosses the threshold the LED color changes.

NXP

# Lab6. Combine FTM and ADC : Blinky FTM ADC

## Things to Learn:

- Co-operating between two modules

Potentiometer

RGB LEDs

## Overview:

- This Example demonstrates that how to realize intended functionality using by running two modules and achieving co operation between both.

- In this simple example, the FTM will function accordingly the value of the ADC port.

## Hardware Connections:

- No Hardware connections required

## Observation:

- The application reads the potentiometer, ADC channel 10.

- According to this value the FTM will adjust its Value register and generated the different blinking frequency and/or different color of LEDs at different ADC value.

# Lab7. Getting Started with Key board Interrupt : KBI

## Things to Learn:

- Initialize KBI



## Overview:

- KBI can be used for HMI Human Machine Interface.
- This example demonstrates that how to configure the port pins in KBI mode so that any change in pin polarity, will generate a KBI interrupt.

## Hardware Connections:
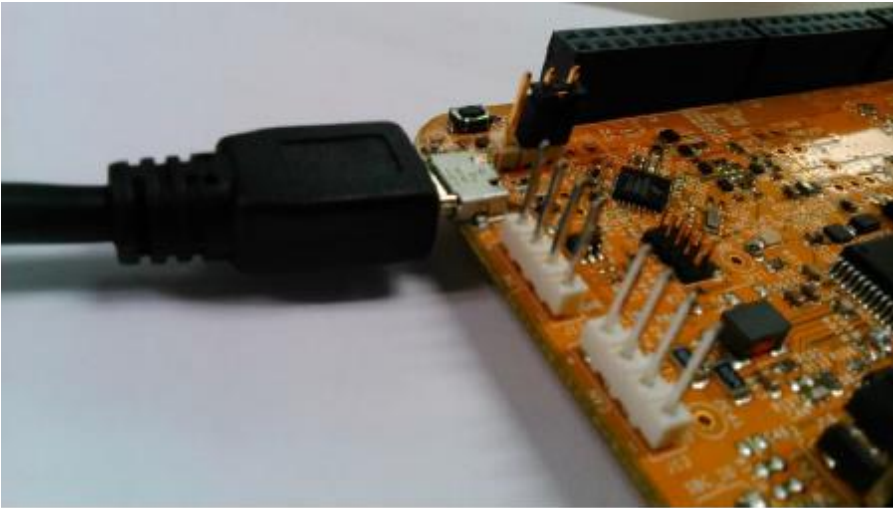
- No Hardware connections required

## Observation:

- Pressing SW2 & SW3 toggles the Green and Red LEDs respectively.

# Lab8. Communicate using Universal Asynchronous Receiver Transmitter (UART)

## Things to Learn:

- Initialize & communicate through UART



**Overview**:

- UART can be used to send and receive data between the different peripherals as well as computer.

**Hardware Connections:**

- Connect Board with computer using micro USB port.
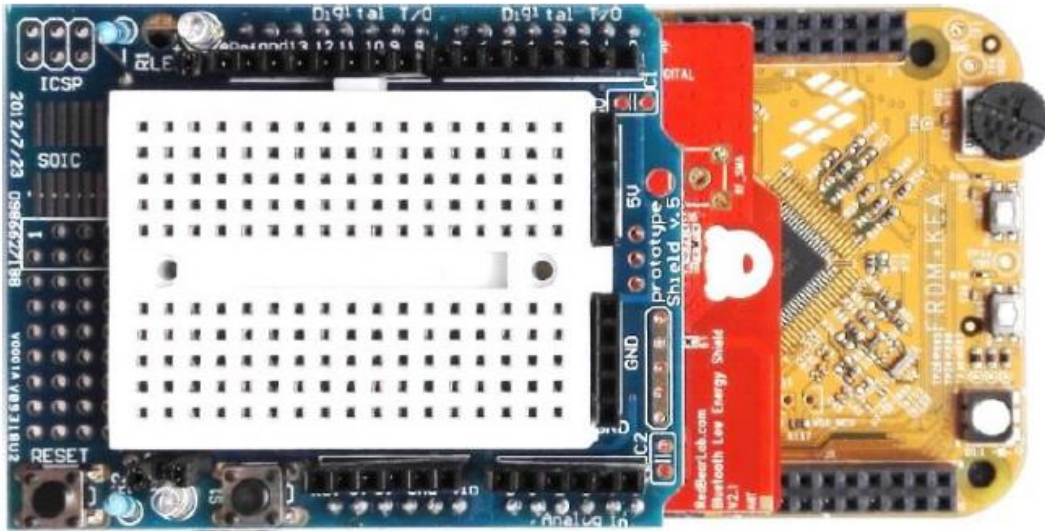- Set your terminal with baud rate of 9600

**Observation:**

- Running the example, displays the message on terminal window.
- Try to write any character on terminal window, board will echo back the same character

# Lab9. Communicate to Your Phone Using Bluetooth

## Things to Learn:

- Expand your board's capability by adding an Arduino shield



**Overview**:

- Bluetooth is a popular wireless communication protocol that is available on most smartphones

## Hardware Connections:

- RedBearLab BLE Shield connected to FRDM KEA following Arduino standard
- Prototype v5 Arduino breadboard stacked atop BLE shield

## Additional Software:

- Free RedBearLab BLE Controller (available in App Store and Play Store)
- Set app to BLE Controller mode and connect to FRDM board

## Observation:

- Change Pin 11, 12, and 13 to PWM mode then use the virtual slides to control LED
- Press SW3 and watch state of Pin 14 change
- Change Pin 22 to Analog mode and turn potentiometer to see reading on app change

# Lab10. Communicate using Local Interconnect Network (LIN)

## Things to Learn:

- Communicate between two FRDM boards through LIN
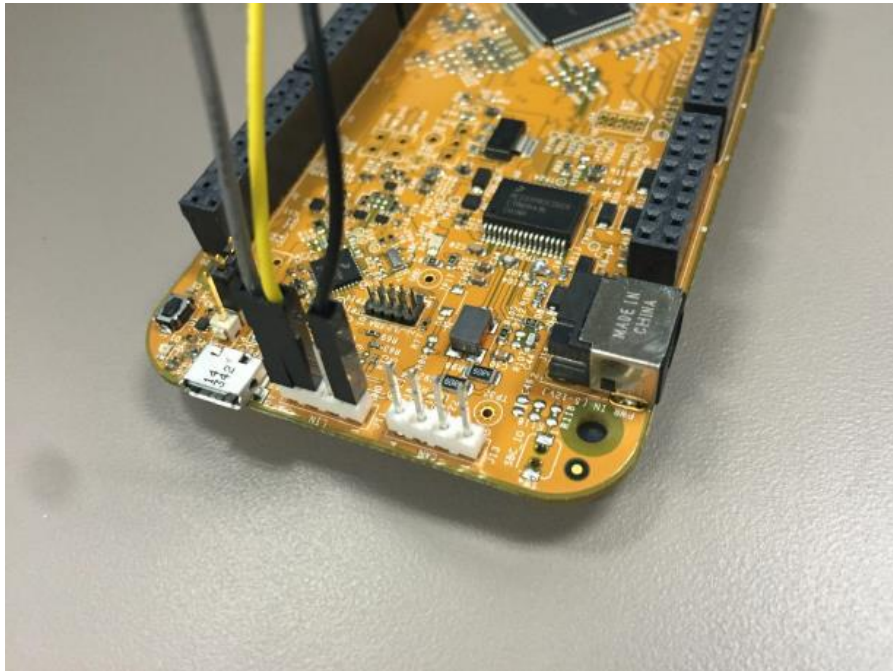


**Overview**:

- LIN is a low cost communication protocol used in automotive applications

**Hardware Connections:**

- Two FRDM KEA Boards
- One 12V Power Supply to J16 of either board
- Connect LIN PHY of both boards using wires: LIN-LIN, Vsup-Vsup , GND-GND
- Enable external power: Connect Pins 1 and 2 of J107 with jumper

**Observation:**

- Potentiometer of Master controls LED of Slave
- Push buttons of Slave controls LED of Master

# Lab11. Control LCD Arduino Shield with KEA

## Things to Learn:

- Communicate between PC, KEA, and LCD Display



**Overview**:

- LCD is a simple way to display images, which may communicate a variety of information from plain messages to interactive outputs such as ADC values.
- Polling Mode: Uses UART polling to communicate with hyperterminal
- Interrupt Mode: Uses UART interrupts

**Hardware Connections:**

- DFRobot DFR0009 Arduino LCD Shield
- Hardware connection to computer using microUSB to USB adapter

**Observation:**

- Hyperterminal software such as Putty
- Set baud rate to 115200

NXP

# Lab11. Control LCD Arduino Shield with KEA



## Observation 1:

- RGB LED will shine blue. It uses the FTM in Edge Aligned PWM mode.

- This code example also uses keyboard interrupts to regulate the duty cycle. Pushing SW2 brightens the LED by increasing the duty cycle while SW3 dims the LED by decreasing it.
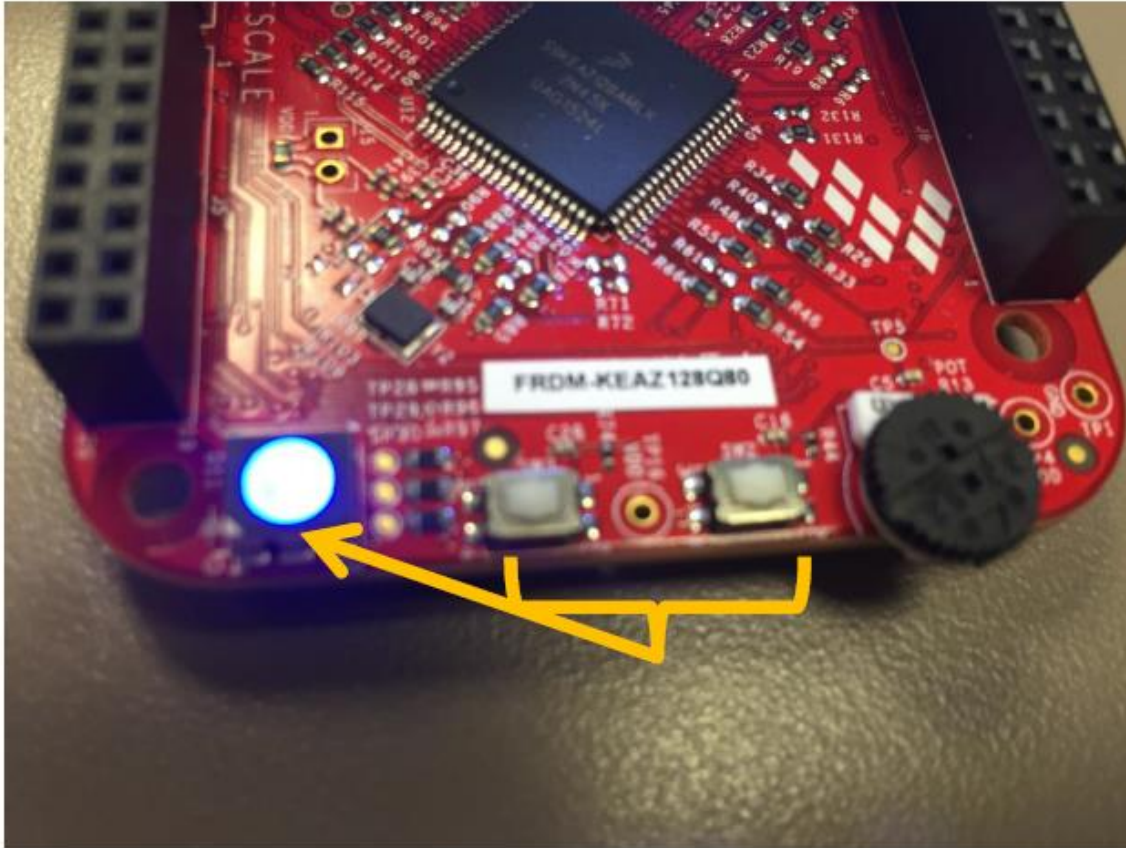
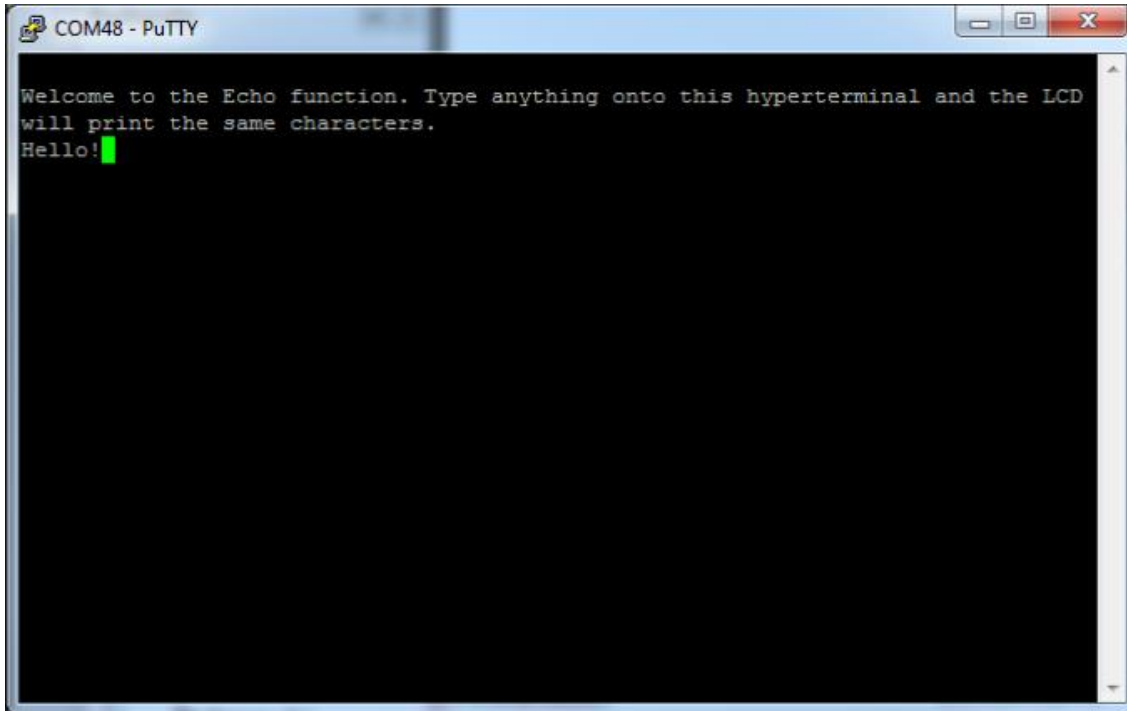# Lab11. Control LCD Arduino Shield with KEA



**Observation 2:**

- On startup, the LCD prints "Welcome!"

- You can then cycle between four LCD functions by pressing the 'Select' button the LCD shield. Demo uses RTC interrupt to time push button checks

- The first function prints a message and continually scrolls the display back and forth to display it in its entirety.
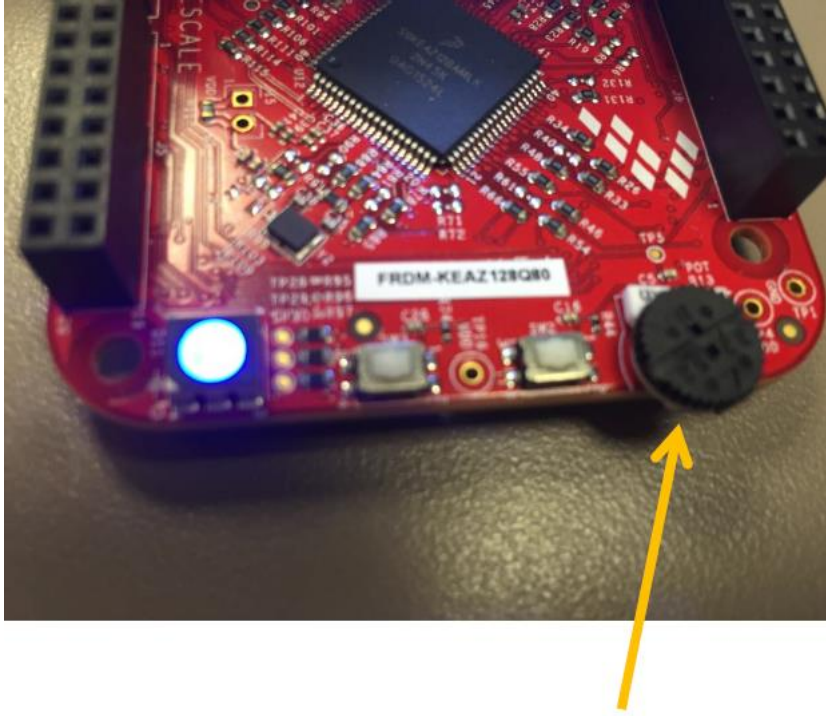
# Lab11. Control LCD Arduino Shield with KEA



**Observation 3:**

- The second LCD function prints on the LCD whatever you type into your hyperterminal

- Communicates with KEA over UART, which forwards the information to the LCD using GPIO

# Lab11. Control LCD Arduino Shield with KEA



**Observation 4:**

- The third LCD function prints the ADC value of the KEA potentiometer

# Lab11. Control LCD Arduino Shield with KEA



## Observation 5:

- The fourth LCD function prints the answer to a simple arithmetic problem of your choosing

- Type into the hyperterminal an arithmetic operation of two operands. Each operand can be any positive integer up to 9999. The LCD will print the answer up to two decimal point precision

- Examples: 4+4, 333*9875, 195/255, 9000-9005

# Lab11. Control LCD Arduino Shield with KEA



## Observation 6:

- The fifth LCD function is a mimic of a magic 8-balls.
- Type a question into the UART terminal (make sure you end the sentence with a '?') and hit *Enter*
- LCD will print you an answer to your question!
- The function uses a random number generator to choose a response.



```
Welcome to the Fortune function. Type any question to the hyperterminal and pres
s Enter. The LCD will display your fortune!
Is this the ooolest demo ever?!?
```

SECURE CONNECTIONS
FOR A SMARTER WORLD