

Android Application Development and Bluetooth Technology

James Cracchiolo

3/28/14

Table of Contents

Introduction	page 3
Objective	page 3
What is Bluetooth?	page 3
What is Android?	page 4
Materials Needed	page 4
Results	page 6
Recommendations	page 6
References	page 6

Introduction

As computer hardware continues to grow, devices such as mobile phones have also gained a vastly improved infrastructure as smart phones are becoming more and more commonplace. Users of such equipment are always trying to find new applications that will benefit them throughout their day to day activities. Many applications are being written send or receive data over communication networks. One such way is to transmit data wirelessly using a Bluetooth configured system where a microcontroller is controlling the input/output of the Bluetooth sensor of the device. The software implementation can be used for Android or iOS devices, but for the use of this application note only the applications that are intended to run using the Android interface will be discussed.

Objective

Instruct an individual how to configure a Bluetooth sensor for connecting to an Android device and writing an Android application that will interface with the Bluetooth sensor.

What is Bluetooth?



Figure 1. Bluetooth trademark

Bluetooth technology allows the ability to connect devices wirelessly to each other. Bluetooth is very useful in connecting devices that are in close proximity to each other and is widely incorporated into many mobile devices today. If a device contains Bluetooth compatibility, it is denoted by the trademark above. Since Bluetooth is easily accessible by the users that own such devices, software applications can be created to control the data that is being transmitted. The range for Bluetooth is meant for shorter distances and is used in setting up a personal area network (PAN) which associates with objects that are within his proximity.

What is Android?



Figure 2. Android trademark

Android is an operating system that runs on mobile devices such as cellular phones and tablets. The operating system features the above trademarked logo thus denoting that it is a device that is running Android. Android is based on the Linux kernel and as such is also open source so it is able to be manipulated. Since the software to write Android applications is also open source, it is free to setup. This is opposite to proprietary software such as Apple's iOS or Microsoft's Windows Phone 8 operating system.

Materials Needed

In order to begin configuring the Bluetooth sensor, the sensor will need to be connected to a microcontroller that will control what data is being sent and received through the sensor. The microcontroller that will be used in this application note is Texas Instruments MSP430 and is shown below.

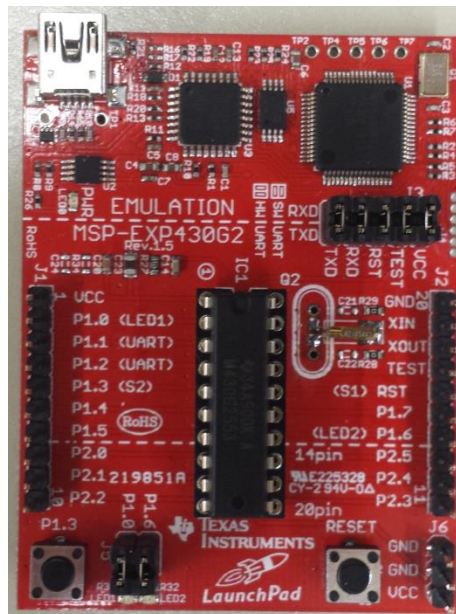


Figure 3. Texas Instruments MSP430 Launchpad

The MSP430 microcontroller contains programmable input/output pins as well as analog-to-digital converters. Programming of the MSP430 is done through Texas Instruments Code Composer Studio. Code Composer Studio uses C code to program the registers, pins, clocks, etc. The pin layout for the MSP430 is the follow figure:

Device Pinout, MSP430G2x13 and MSP430G2x53, 20-Pin Devices, TSSOP and PDIP

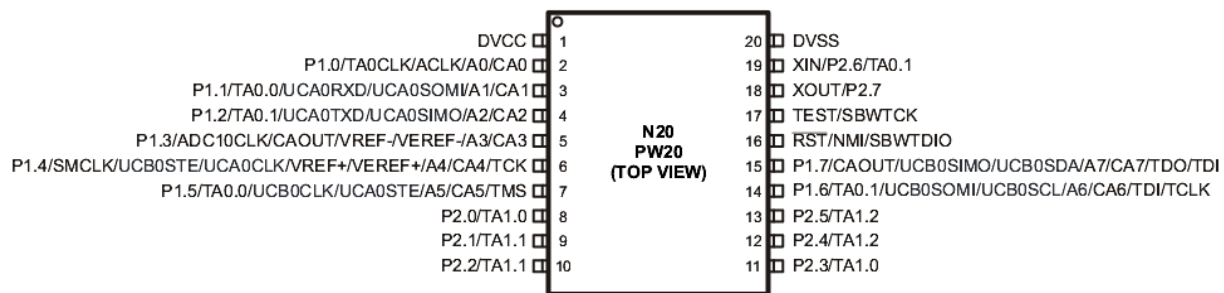


Figure 4. MSP430 Pin Layout

The specific registers and pins that are used in the code can be found in the TI datasheet specific to the MSP430. Some sample code to configure a pin for output is the following:

```
#include <msp430g2553.h>
unsigned int delay;

void main(void)
{
    // turns of the Watchdog Timer
    WDTCTL = WDTPW + WDTHOLD;

    // sets the green led
    P1DIR = 0x40;
    while(1)
    {
        // makes the green LED flash
        P1OUT ^= 0x40;
        // delay for green LED
        for(delay=0; delay<50000;delay++);
    }
}
```

Figure 5. Code Composer Studio Example Code

This code is configuring P1.6 of the microcontroller which is connected to the green LED on the launch pad. The include at the top is necessary to access the libraries used by the MSP430. This code turns the green LED on and shows how to write a delay loop which will be necessary when

refreshing data. This code is meant as an example as to how to setup the input/output pins of the MSP430 as the Bluetooth sensor will need to be configured in a similar way.

For the Android application development, everything can be downloaded by going to Androids website and downloading the developer's Android SDK package. After installing the Android package, it is also necessary to have an Android device to test the application.

Results

The Android application should be able to achieve connectivity to the Bluetooth sensor and be able to communicate with the device that is associated with it. The user can send data to the Bluetooth device using an Android phone or tablet. The Android device can also act as a receiver and be used to monitor the values that are controlled by the microcontroller. For example, with the sample code for the microcontroller that turns an LED on and off, an Android application can be written to control that functionality.

Recommendations

A different type of microcontroller can also be used. One such microcontroller is the Arduino Uno which uses a different coding environment than the MSP430 therefore offering users with another option to rely upon.

References

The code sample is from the ECE480 lab notes