ASEE'S VIRTUAL CONFERENCE

At Home with Engineering Education

JUNE 22 - 26, 2020 #ASEEVC

Paper ID #29072

Programmable System-On-Chip (PSoC) Usage in Embedded Programming Courses

Prof. David R. Loker, Penn State Erie, The Behrend College

David R. Loker received the M.S.E.E. degree from Syracuse University in 1986. In 1984, he joined General Electric (GE) Company, AESD, as a design engineer. In 1988, he joined the faculty at Penn State Erie, The Behrend College. In 2007, he became the Chair of the Electrical and Computer Engineering Technology Program. His research interests include wireless sensor networks, data acquisition systems, and communications systems.

Mr. Stephen A. Strom, Pennsylvania State University, Behrend College

Stephen Strom is a lecturer in the Electrical and Computer Engineering Technology department of Penn State Behrend, and holds a B.S. in electrical engineering from Carnegie Mellon University. His career includes over thirty years experience in designing and programming embedded systems and has multiple patents for both hardware designs and software algorithms

Programmable System-On-Chip (PSoC)
Usage in Embedded Programming Courses

Abstract

Part of the requirements for an Electrical and Computer Engineering Technology (ECET) program includes the ability for students to design and implement embedded systems in a variety of courses. Typical courses can include C programming, microcontrollers, instrumentation and measurement systems, wireless communications systems, networking, and control systems, and they can be lower-division and/or upper-division courses. Lab content within these courses provides students with hands-on projects which are intended to support the lecture material. The intent of this paper is to demonstrate that an inexpensive Programmable System-On-Chip embedded board (PSoC 5LP) can be used as a basis for many lab projects within a wireless communications systems course. The software development environment for the PSoC provides the flexibility to program in C as well as to use the schematic entry capability. The goal of this paper is to introduce a series of labs that can be used within this course. Possible topics for lab projects include: analog-to-digital conversion, digital-to-analog conversion, sensor data collection, timers, serial communication, serial peripheral interface (SPI) protocol, inter-integrated circuit (I^2C) protocol, on-off keying (OOK) modulation, RF transceiver (e.g., ZigBee, Bluetooth, WiFi) interfacing, and remote control. For lab projects, documentation (e.g., engineering requirements, schematics, PSoC device configurations, and 'C' code) is provided. Additionally, a summary and assessment are provided.

Introduction

In an Electrical and Computer Engineering Technology (ECET) program, there are many courses that use embedded systems as part of the objectives. The format for these courses is to teach 'C' or 'C++' programming, with embedded hardware and software, as either a core component to the course or as a means of implementing various applications. For example, in a wireless communications systems course, an embedded system can be used as the primary means for remote sensor data collection and transmission.

The goal of this paper is to show how an inexpensive embedded device (Programmable System-On-Chip embedded board – PSoC 5LP) can be used in a wireless communications systems course[1].

Course Overview

The wireless communications systems course is a junior-level course in the Electrical and Computer Engineering Technology program. The course provides foundational material in spectrum analysis, filtering, serial communications, and analog and digital modulation and demodulation. Applications include wireless networking for the development of personal area networks (PAN) and local area networks (LAN). Prerequisite material includes LabVIEW for the design of measurement systems[3-6]. Additionally, students complete a C++ course on high-level programming with procedure-oriented and object-oriented applications for electrical and electronic problem solving.

The wireless communications systems course is 3 credits and consists of 2 hours of lecture and 2 hours of lab per week. The lab content is designed to emphasize topics discussed during lecture. Each lab is considered a design project, since it contains a series of engineering design requirements and takes either 1 or 2 weeks to complete, depending upon the scope of the project. Each project is completed by a student team which consists of either 1 or 2 students (some students prefer to work by themselves). Students pick their team members at the beginning of the semester.

For all of the projects, students are expected to work outside of the scheduled lab time in order to complete the design objectives. A listing of the lab projects for the course is shown in Table 1. The first four lab projects utilize LabVIEW with the myDAQ, while most of the remaining labs utilize the PSoC 5LP.

Table 1. Lab Project Schedule.

Lab #	**Lab Title**	**# of weeks to complete**
Lab 1	Signal Generation and Spectrum Analysis	1 week
Lab 2	Voice Data Collection	1 week
Lab 3	Passive Filter Design and Analysis	1 week
Lab 4	Digital Graphic Equalizer Design	1 week
Lab 5	Temperature Data Collection using PSoC 5LP	1 week
Lab 6	Audio Data Collection using PSoC 5LP	1 week
Lab 7	Serial Communication using PSoC 5LP	1 week
Lab 8	Infrared Communications System	1 week
Lab 9	Wireless Sensor Network using XBee RF Modules	2 weeks
Lab 10	Remote Control System using Bluetooth Modules	2 weeks
Lab 11	Web Server using WiFi Modules	2 weeks

PSoC 5LP Board/Development Environment

An overview of the PSoC 5LP board and development environment is contained within this paper[2]. The PSoC 5LP board is a re-configurable gate array that includes both analog and digital peripheral circuits. Users can select from a variety of components.

- Logic gates
- UART devices
- SPI devices
- ADC devices
- DAC devices
- Opamp devices
- Analog MUX

The wide variety of analog and digital components are available for usage in lab projects for a wireless communications systems course.

Lab Projects:

Possible lab projects for a wireless communications systems course include: analog-to-digital conversion, digital-to-analog conversion, sensor data collection, timers, serial communication, serial peripheral interface (SPI) protocol, inter-integrated circuit (I^2C) protocol, on-off keying (OOK) modulation, RF transceiver (e.g., ZigBee, Bluetooth, WiFi) interfacing, and remote control. The goal for this paper is to discuss the following labs:

1. Temperature Data Collection using PSoC 5LP – The objectives are to utilize a temperature sensor to read voltage every 1000mS, convert the voltage to temperature, display the temperature on an LCD display, and transmit the temperature to a PC through the UART interface.
2. Audio Data Collection using PSoC 5LP – The objectives are to collect audio data using an Electret microphone breakout board, transmit the data to a PC for storage to a text file, and develop a LabVIEW software program for playback through the PC sound card.
3. Infrared Communications System – The objective is to design a simplex IR communication system to transmit temperature from the PSoC 5LP to a PC.
4. Remote Control System using Bluetooth Modules – The objective is to design a remote temperature sensor control system using Roving Networks RN-41 Bluetooth modules.

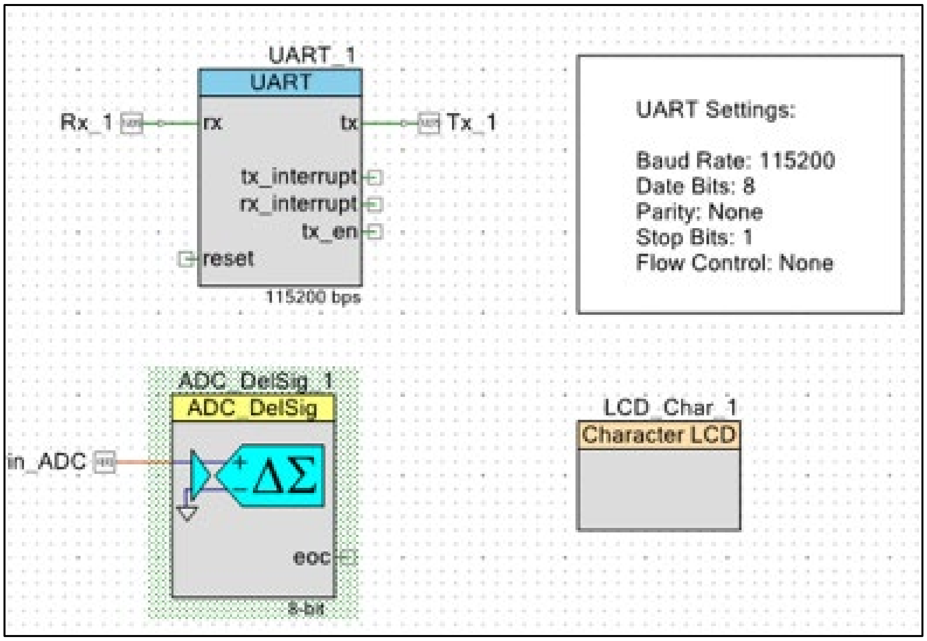
Temperature Data Collection using PSoC 5LP

The engineering requirements for this lab project are listed below.

Engineering Requirements

- LM35 used as the temperature sensor
- Delta-Sigma ADC component on the PSoC 5LP used for reading sensor voltage
- Character LCD component on the PSoC 5LP for displaying temperature
- UART component on the PSoC 5LP used for transmitting temperature to the PC

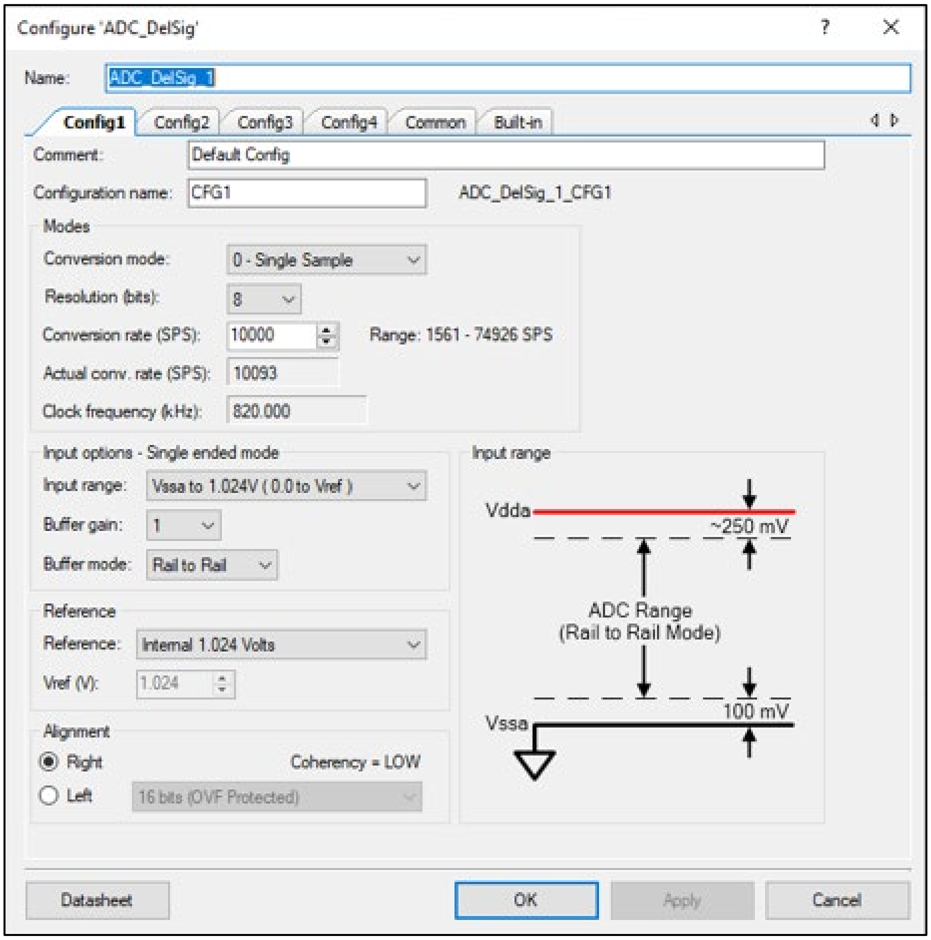
The schematic, ADC configuration, wire mapping, and main ‘C’ code are shown in Figures 1 - 4, respectively. The schematic consists of a UART, delta-sigma ADC, and Character LCD display. The ADC is configured for collecting a single sample with 8 bits of resolution, a full-scale voltage of 1.024V, and single-ended mode. The LCD display is connected to Port 2 (pins 0 through 6) of the PSoC 5LP. The ‘C’ code is designed to collect one voltage from the LM35 temperature sensor every 1000mS, convert the voltage to temperature in degrees C, transmit the temperature through the UART to the PC, and display the temperature on the Character display.



UART Settings:

Baud Rate: 115200
Date Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

Figure 1 – Temperature Data Collection Schematic



Name: ADC_DelSig_1

Comment: Default Config

Configuration name: CFG1 ADC_DelSig_1_CFG1

Conversion mode: 0 - Single Sample

Resolution (bits): 8

Conversion rate (SPS): 10000 Range: 1561 - 74926 SPS

Actual conv. rate (SPS): 10093

Clock frequency (kHz): 820.000

Input range: Vssa to 1.024V (0.0 to Vref)

Buffer gain: 1

Buffer mode: Rail to Rail

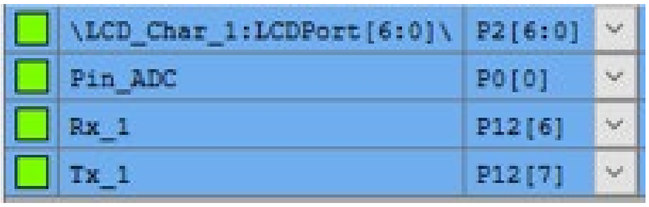
Reference: Internal 1.024 Volts

Vref (V): 1.024

● Right Coherency = LOW

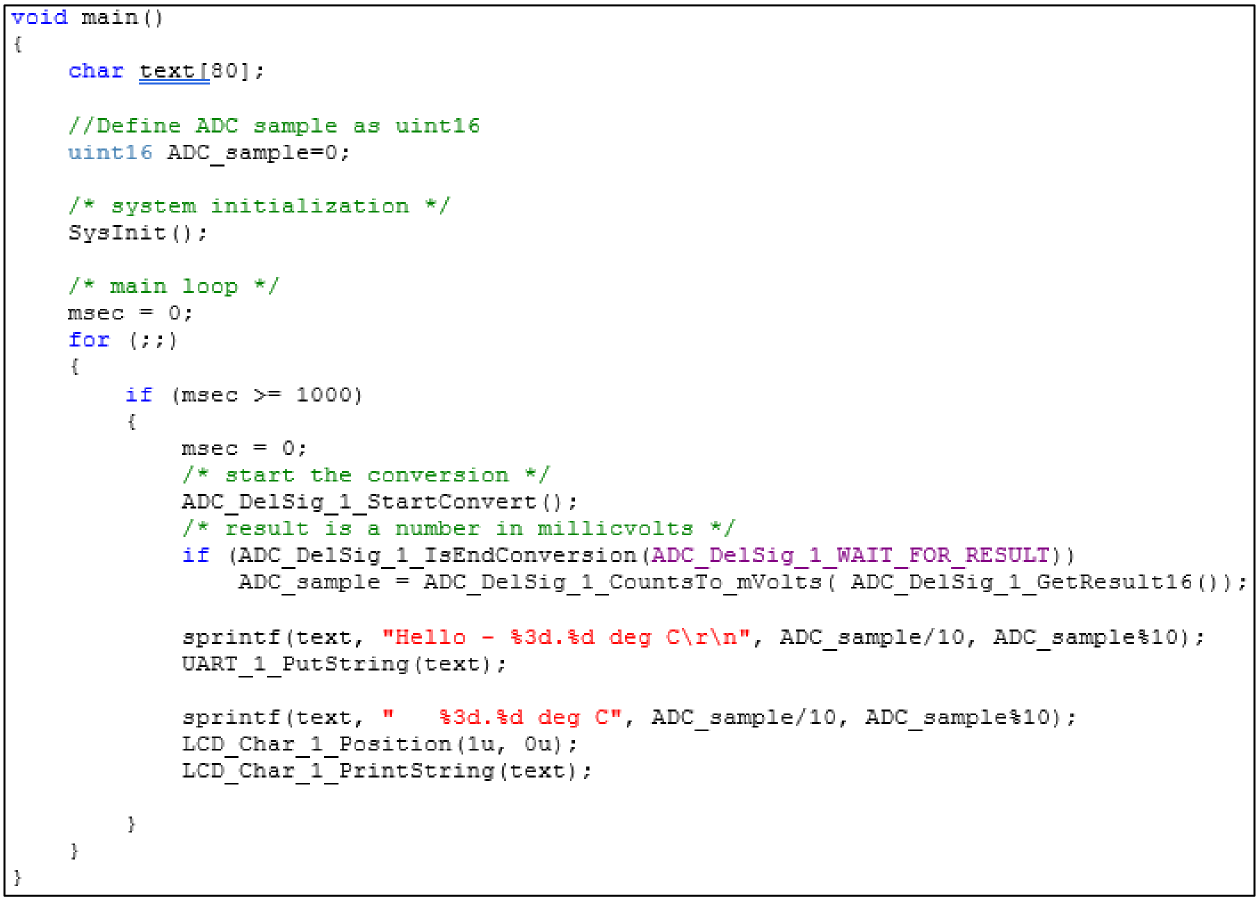
○ Left 16 bits (OVF Protected)

Figure 2 – ADC Configuration for Temperature Data Collection



🟩	\LCD_Char_1:LCDPort[6:0]\	P2[6:0]	⌄
🟩	Pin_ADC	P0[0]	⌄
🟩	Rx_1	P12[6]	⌄
🟩	Tx_1	P12[7]	⌄

Figure 3 – Wire Mapping for Temperature Data Collection



```
void main()
{
    char text[80];

    //Define ADC sample as uint16
    uint16 ADC_sample=0;

    /* system initialization */
    SysInit();

    /* main loop */
    msec = 0;
    for (;;)
    {
        if (msec >= 1000)
        {
            msec = 0;
            /* start the conversion */
            ADC_DelSig_1_StartConvert();
            /* result is a number in millicvolts */
            if (ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT))
                ADC_sample = ADC_DelSig_1_CountsTo_mVolts( ADC_DelSig_1_GetResult16());

            sprintf(text, "Hello - %3d.%d deg C\r\n", ADC_sample/10, ADC_sample%10);
            UART_1_PutString(text);

            sprintf(text, "   %3d.%d deg C", ADC_sample/10, ADC_sample%10);
            LCD_Char_1_Position(1u, 0u);
            LCD_Char_1_PrintString(text);

        }
    }
}
```

Figure 4 – Main ‘C’ Code for Temperature Data Collection

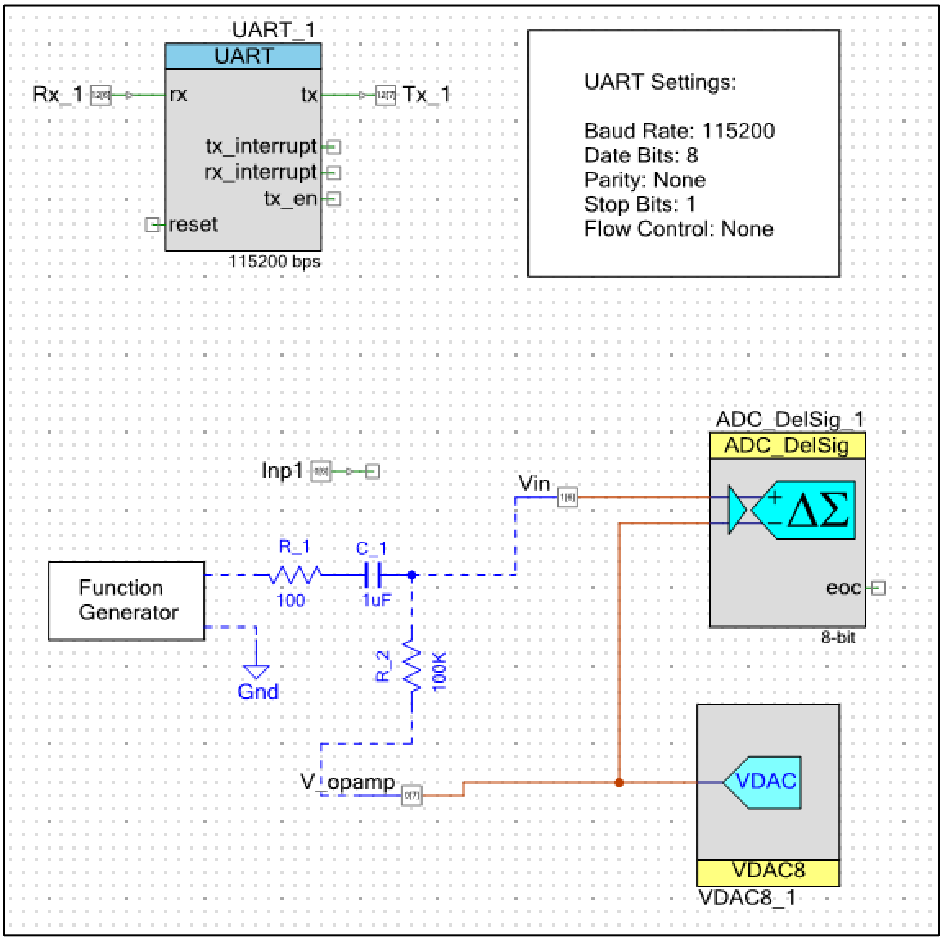
Audio Data Collection using PSoC 5LP

The engineering requirements for this lab project are listed below.

Engineering Requirements

- Electret microphone breakout board used for audio sensing
- Delta-Sigma ADC component on the PSoC 5LP used for reading sensor voltage
- UART component on the PSoC 5LP used for transmitting temperature to the PC
- PC terminal program used to capture audio in a text file
- LabVIEW used to read the text file and drive PC the sound card

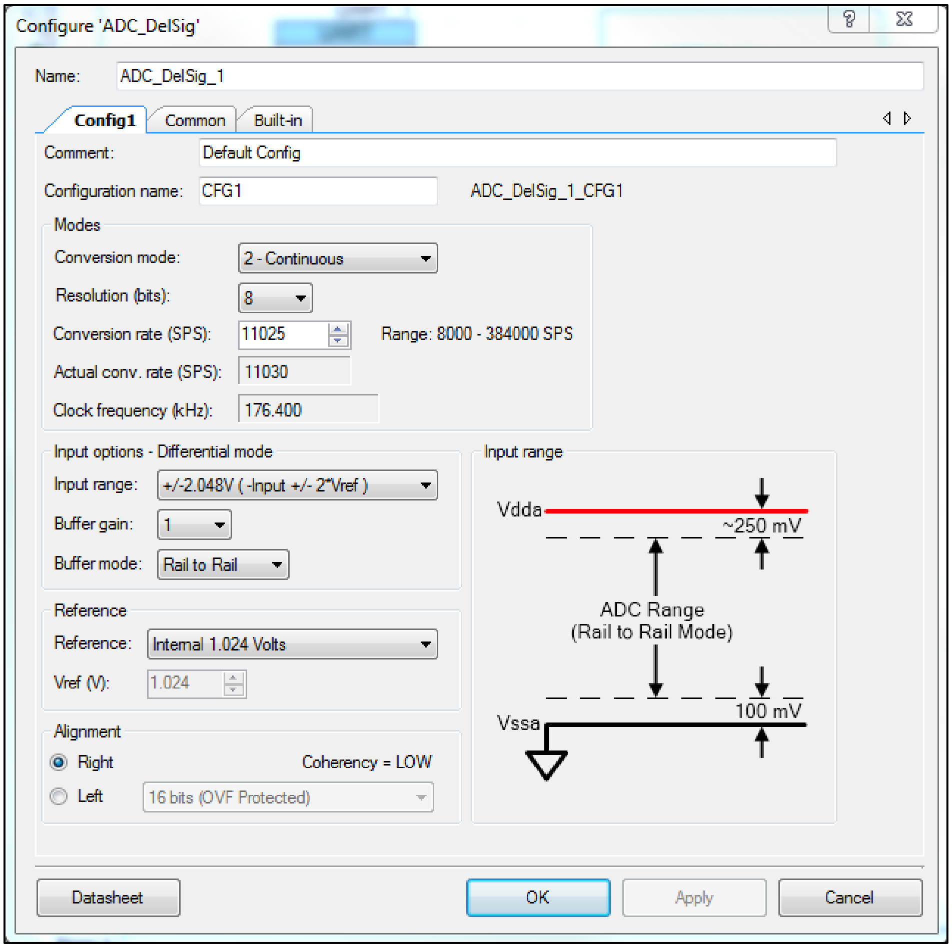
The schematic, ADC configuration, main 'C' code, and LabVIEW block diagram are shown in Figures 5 - 8, respectively. The schematic consists of a UART, delta-sigma ADC, and VDAC (voltage DAC). A function generator is used for initial testing, and then it is replaced by the electret microphone breakout board[8]. External components (resistors and capacitor) are used to remove the DC offset from the breakout board and to add a 2V DC offset (from the VDAC) for the reference voltage of the ADC. The ADC is configured for continuously collecting voltages from the breakout board at a rate of 11025 Hz with 8 bits of resolution, and with an input voltage range of ±2.048V with respect to the reference voltage. The 'C' code is designed to collect 11025 voltage samples (for one second) and transmit the results through the UART to the PC. The LabVIEW program on the PC is designed to scale the data to drive the PC sound card.



UART Settings:

Baud Rate: 115200
Date Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

Figure 5 – Audio Data Collection Schematic



Configure 'ADC_DelSig'

Name: ADC_DelSig_1

Comment: Default Config

Configuration name: CFG1 ADC_DelSig_1_CFG1

Modes

Conversion mode: 2 - Continuous

Resolution (bits): 8

Conversion rate (SPS): 11025 Range: 8000 - 384000 SPS

Actual conv. rate (SPS): 11030

Clock frequency (kHz): 176.400

Input options - Differential mode

Input range: +/-2.048V (-Input +/- 2*Vref)

Buffer gain: 1

Buffer mode: Rail to Rail

Reference

Reference: Internal 1.024 Volts

Vref (V): 1.024

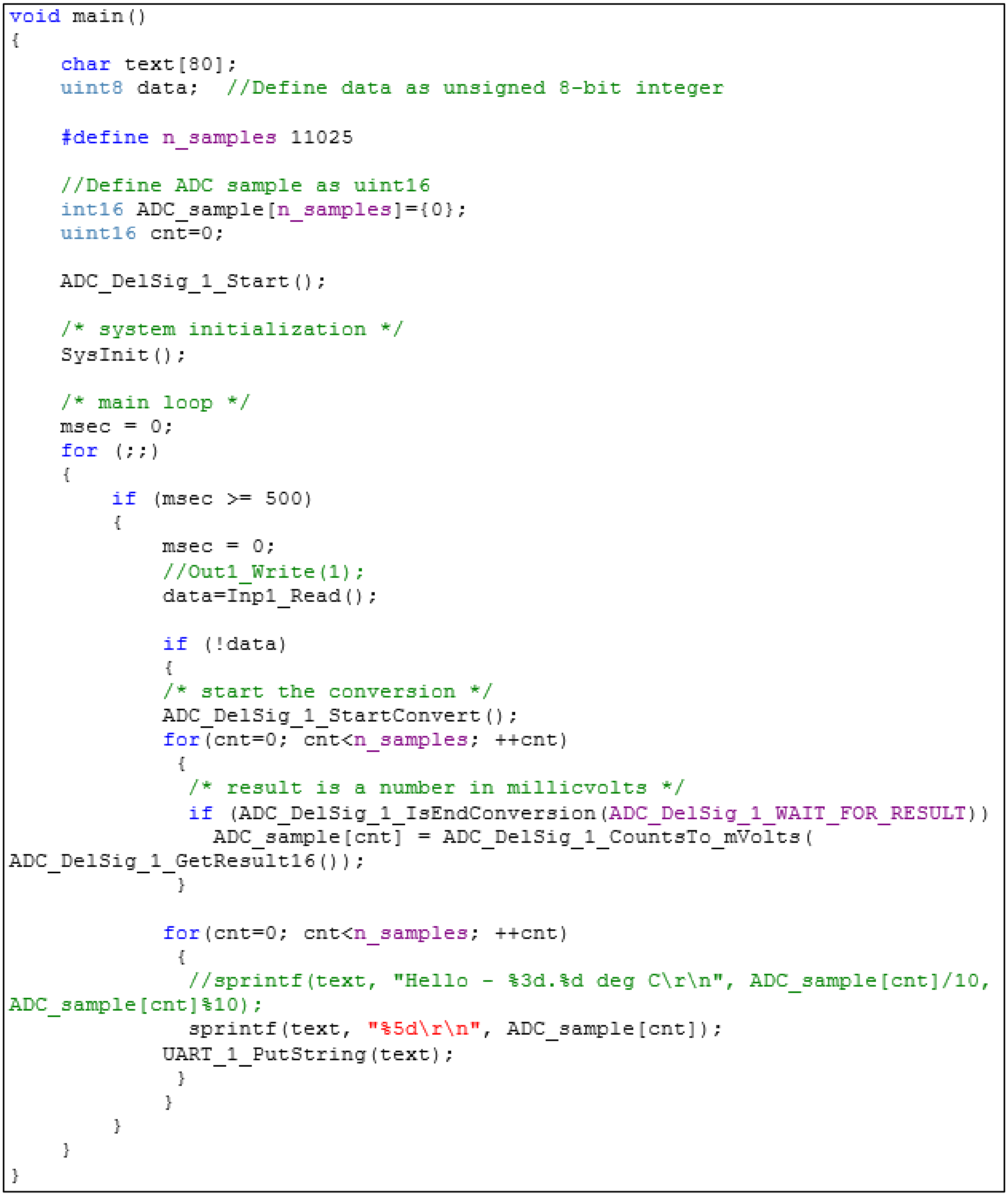
Alignment

● Right Coherency = LOW

○ Left 16 bits (OVF Protected)

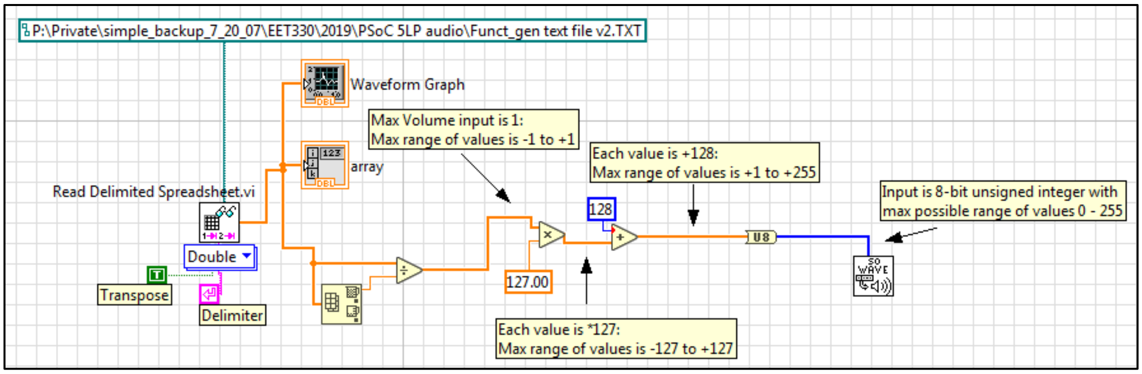
Input range

Figure 6 – ADC Configuration for Audio Data Collection



```c
void main()
{
    char text[80];
    uint8 data;  //Define data as unsigned 8-bit integer

    #define n_samples 11025

    //Define ADC sample as uint16
    int16 ADC_sample[n_samples]={0};
    uint16 cnt=0;

    ADC_DelSig_1_Start();

    /* system initialization */
    SysInit();

    /* main loop */
    msec = 0;
    for (;;)
    {
        if (msec >= 500)
        {
            msec = 0;
            //Out1_Write(1);
            data=Inp1_Read();

            if (!data)
            {
            /* start the conversion */
            ADC_DelSig_1_StartConvert();
            for(cnt=0; cnt<n_samples; ++cnt)
             {
              /* result is a number in millicvolts */
              if (ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT))
                ADC_sample[cnt] = ADC_DelSig_1_CountsTo_mVolts(
ADC_DelSig_1_GetResult16());
             }

            for(cnt=0; cnt<n_samples; ++cnt)
             {
              //sprintf(text, "Hello - %3d.%d deg C\r\n", ADC_sample[cnt]/10,
ADC_sample[cnt]%10);
              sprintf(text, "%5d\r\n", ADC_sample[cnt]);
            UART_1_PutString(text);
             }
            }
        }
    }
}
```

Figure 7 – Main ‘C’ Code for Audio Data Collection



P:\Private\simple_backup_7_20_07\EET330\2019\PSoC 5LP audio\Funct_gen text file v2.TXT
Waveform Graph
array
Read Delimited Spreadsheet.vi
Double
Transpose
Delimiter
127.00
128
U8

Max Volume input is 1:
Max range of values is -1 to +1

Each value is +128:
Max range of values is +1 to +255

Input is 8-bit unsigned integer with
max possible range of values 0 - 255

Each value is *127:
Max range of values is -127 to +127

Figure 8 – LabVIEW Program for Audio Data Collection

Infrared (IR) Communications System

The engineering requirements for this lab project are listed below.

Engineering Requirements

- LM35 used as the temperature sensor
- Delta-Sigma ADC component on the PSoC 5LP used for reading sensor voltage
- PWM component on the PSoC 5LP used for setting the carrier frequency (38KHz +/- 5%) and duty cycle (10% to 25%)
- UART component on the PSoC 5LP connected to the ULN2003A (external high-current darlington transistor array IC) and IR LED (OPTEK 295A) for driving peak LED current (300mA to 400mA)
- External VISHAY IR Receiver Module (TSOP34838) for reading the IR signal
- External TTL to RS232 level converter IC (ICL3232) for connecting the signal to the PC UART
- LabVIEW program used to read the read the data from the receive buffer of the PC UART and to display temperature

The system block diagram is shown in Figure 9. The system block diagram consists of an IR transmitter and IR receiver for simplex communication between the PSoC 5LP and a PC. The transmitter is designed to produce a carrier waveform which alternates between a carrier frequency and an off (continuously low) state. This results in an on-off key (OOK) modulated waveform. Modulation has the advantage of minimizing noise from ambient light and other noise sources detected within the receiver. A darlington transistor IC (ULN2003A) is used to drive sufficient peak current through the IR LED for adequate transmission distances. For the IR receiver, an IR receiver module is used to demodulate the signal and an RS232 level converter is used to convert the signal to a voltage suitable for the PC UART.

The PSoC 5LP schematic (Figure 10) consists of a UART, PWM device, and an ADC. The ADC is designed to read the voltage from the LM35 temperature sensor, convert the voltage to temperature in degrees C, and transmit the temperature through the UART. The PWM device is designed to oscillate at the carrier frequency and duty cycle (Figure 11). The UART is connected to the PWM 'kill' input such that an OOK waveform is produced. A LabVIEW terminal program (Figure 12) is designed to read the data from the receive buffer of the PC serial port and display the results on the front panel.

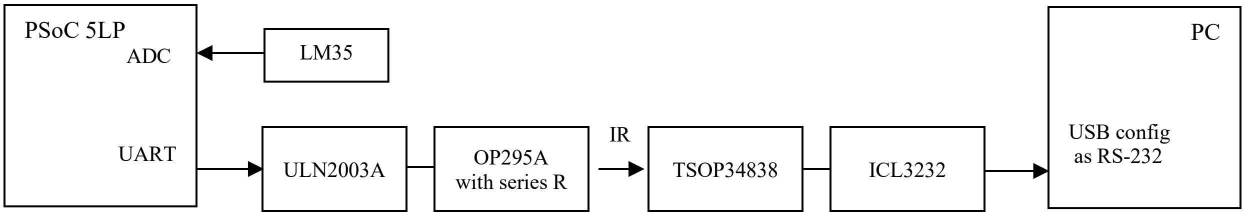
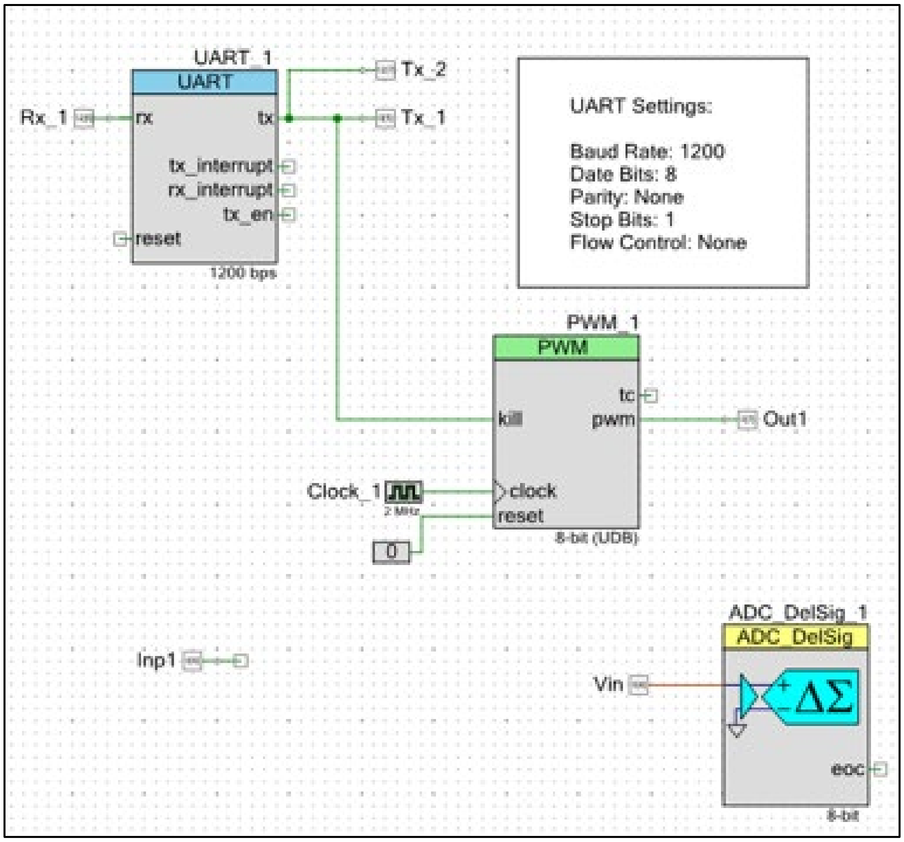


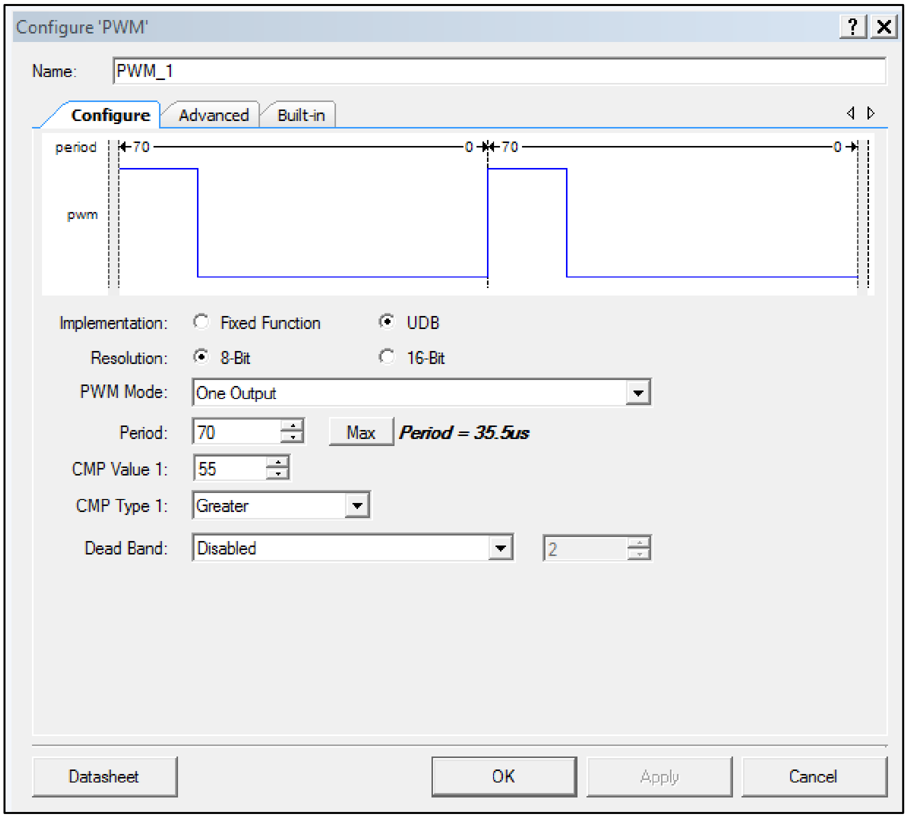
Figure 9 – Block Diagram for IR Communications System



UART Settings:

Baud Rate: 1200
Date Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

Figure 10 – IR Communications System Schematic



Name: PWM_1

Implementation: ○ Fixed Function ● UDB

Resolution: ● 8-Bit ○ 16-Bit

PWM Mode: One Output

Period: 70 *Period = 35.5us*

CMP Value 1: 55

CMP Type 1: Greater

Dead Band: Disabled 2

Figure 11 – PWM Configuration for IR Communications System

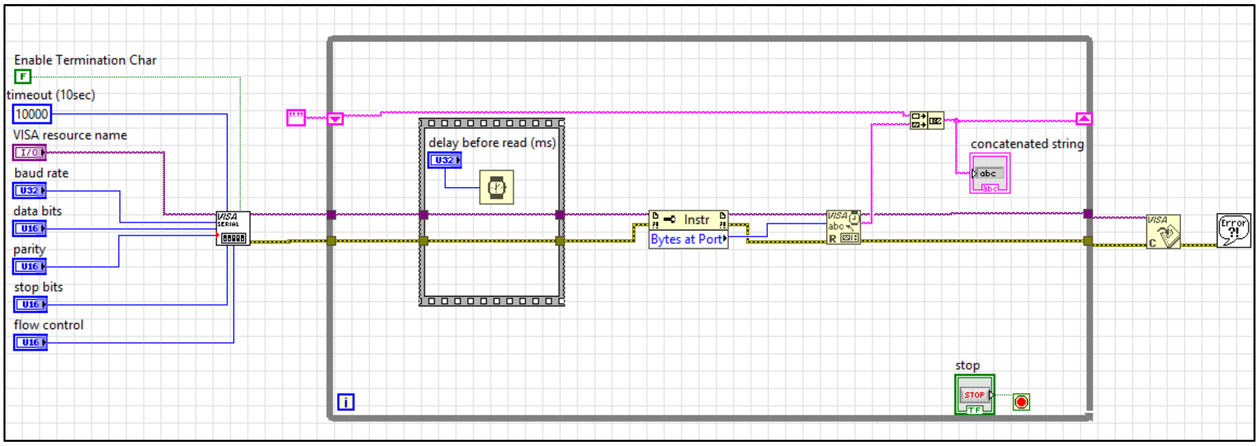


Figure 12 – LabVIEW Program for IR Communications System

Remote Control System using Bluetooth Modules

The engineering requirements for this lab project are listed below.

Engineering Requirements

- PSoC software:
    - Continuously reads the UART until a “start collection” command (character) is received. This will begin the temperature collection.
    - LM35 is used as the temperature sensor.

    - One voltage sample (representing temperature) is collected every 500mS for a total of 5 seconds (10 voltage samples).
    - Each voltage sample is converted to temperature in °C and transmitted.
    - The above process is repeated if another “start collection” command is received.
- LabVIEW software:
    - A state machine is designed.
    - Automatically establishes connection to PSoC Bluetooth module (RN-41).
    - Waits for the user to depress the “start collection” command button on the front panel. This sends a character (1-byte) to the PSoC Bluetooth module.
    - Waits to receive the temperature data, display temperatures in °C in an indicator on the front panel, and saves temperature to a file on disk.
    - An email containing temperatures in °C is automatically sent to your email address.
    - Unless a stop button has been depressed, program re-loops to wait for the user to depress the “start collection” command button.

Text commands are used to configure each Bluetooth module[9]. Each command can be entered in either upper or lower case. Arguments for commands are delimited by using a comma. To enter the command mode, the characters “$$$” are typed. The module will respond with “CMD”. To exit the command mode, “---<cr>” is typed, where <cr> is the enter button. The device will respond with “END”. Valid commands are returned with an “AOK” response, and invalid ones are returned with “ERR”. These commands are used to configure each device:

• SF,1<cr>	set to factory default
• ST,255<cr>	enables device to be configured
• SM,0<cr>	slave mode (devices can discover and connect to this device)
• SA,0<cr>	disable authentication
• R,1<cr>	reboot

The system block diagram is shown in Figure 13. The system block diagram consists of two Bluetooth RF transceivers, an LM35 temperature sensor, a PSoC 5LP, and a PC for remote sensor control. A state machine is designed for the LabVIEW PC program. Table 2 lists each state and function for the state machine. Figure 14 shows the LabVIEW block diagram for the CommandMode state.

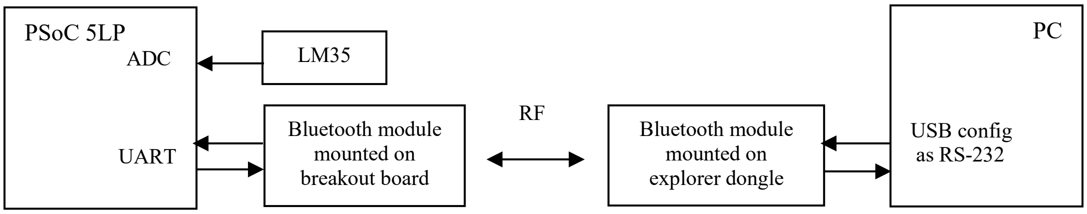

PSoC 5LP
ADC
UART
LM35
Bluetooth module mounted on breakout board
RF
Bluetooth module mounted on explorer dongle
PC
USB config as RS-232

Figure 13 – Block Diagram for Remote Control System

Table 2. LabVIEW State Machine for Remote Control System

State	**Function**
CommandMode	Sends characters to enter command mode.
WaitforCommand	Delay for 750 mS to allow module to enter command mode.
Connect	Sends command for local Bluetooth module to connect to remote Bluetooth module.
WaitforConn	Delay for 5000 mS to allow modules to establish a communications link.
Command	Waits for user to depress command button.
XMT	Transmits to temperature command to initiate the temperature data collection.
Wait	Delay for 10,000 mS to allow temperatures to be collected.
Read	Reads all temperature samples
Save	Save all temperature samples to a text file.
email	Emails all temperature samples.
Stop	Sends characters to enter command mode, delays for 750 mS to allow module to enter command mode, and then kills communication link between modules.

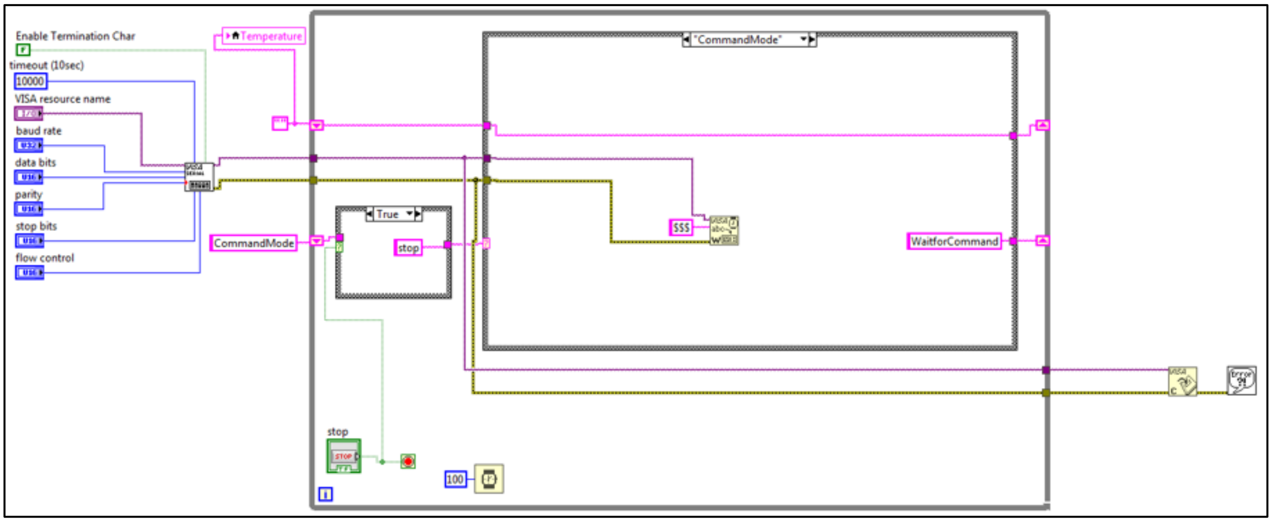


Figure 14 – LabVIEW State Machine for Remote Control System with “CommandMode” State

Summary and Assessment

The PSoC 5LP is an inexpensive, 32-bit processor board that can be used for a variety of embedded projects, spanning from simple to complex systems, and can be applied across a variety of courses. It contains a programmable digital and analog logic array, with numerous onboard components. The software development environment (PSoC Creator) is a free download which makes it inexpensive to use. The download also installs datasheets for the components and project examples which further simplifies classroom instruction.

The projects presented in this paper are intended for a wireless communications systems course, and are only a subset of the various projects students complete during the course. During the 5th week of the semester, students begin using the PSoC 5LP, and a 'tutorial' software project is provided for download to the students. This is used as a template for the design projects for the labs. This template implements a software timer that transmits the text 'HELLO' every 1000mS through the UART output such that it can be displayed on the PC using a terminal program. Additionally, a tutorial text document is provided to the students that shows them how to use the PSoC Creator software development environment, and how to compile and download the 'tutorial' program to the PSoC 5LP. This document also provides instructions on how to use various components (e.g., Digital I/O, ADC, MUX, Character LCD Display, SPI, etc.).

Using the tutorial text document and software download, students learn the software development environment quickly. They can efficiently use their knowledge to create new designs and can complete the lab projects with minimal instruction. Student feedback regarding the PSoC 5LP has been favorable overall. The schematic entry and device configuration are particularly appealing for electrical students who are not as proficient in 'C' code development.

Bibliography

1. Web Site http://www.cypress.com/documentation/development-kitsboards/cy8ckit-059-psoc-5lp-prototyping-kit-onboard-programmer-and
2. Strom, S. A., & Loker, D. "Programmable System-On-Chip (PSoC) Usage in an Engineering Technology Program." *Annual Meeting, American Society for Engineering Education.* (2016).
3. Bishop, Robert H., *Learning with LabVIEW 2009*, Pearson Education, 2010.
4. Travis, Jeffrey and Jim Kring, *LabVIEW for Everyone*, 3rd Edition, Pearson Education, 2007.
5. Essick, John, *Hands-On Introduction to LabVIEW for Scientists and Engineers*, 2nd Edition, Oxford University Press, 2013.
6. Loker, D., & Strom, S. A. "Automated Test & Measurement System for a Power Supply and Control Board." *Annual Meeting, American Society for Engineering Education.* (2016).
7. Web Site https://www.cypress.com/products/psoc-creator-integrated-design-environment-ide
8. Web Site https://www.sparkfun.com/products/12758
9. Web Site https://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-bluetooth-um.pdf