S5D9 Lab ADC MIC
By
Michael Li
(2/5/2018)
https://www.miketechuniverse.com

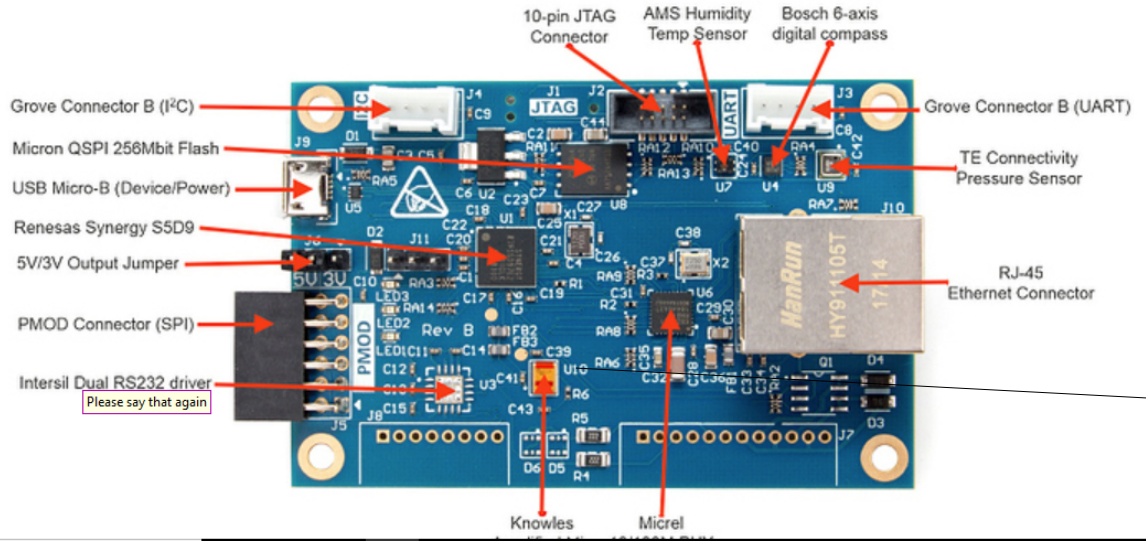
E2 Studio 5.4.0.023
SSP 1.3.0

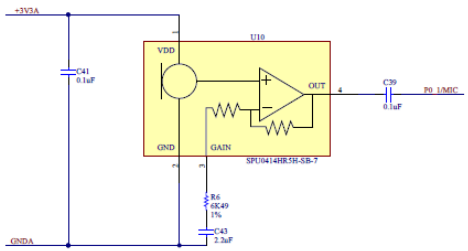
P0_1 ADC0

ADC input is connected to the on board microphone. This example will read from ADC to interpret the sound loudness level.

Renesas Synergy Platform S5D9 IoT Fast Prototyping Kit (product page 2)

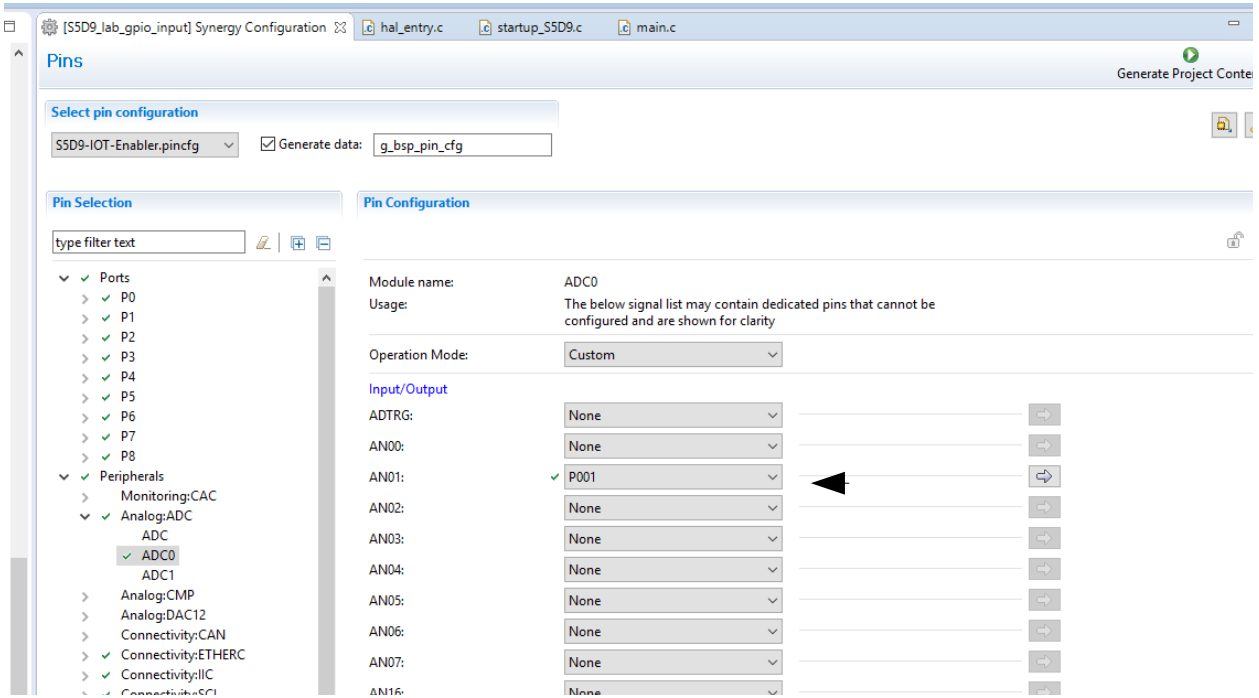
- Synergy S5D9 MCU with ARM CM4F @120MHz, 2M Flash and 640KB SDRAM
- External 256Mbits serial Nor QSPI flash for extra data and application storage
- Integrated acoustic, motion, pressure, temperature and humidity sensors
- 10/100Base-T Ethernet port for wireline connectivity to cloud
- USB 2.0 full speed as device and 5V power input
- Three colored LEDs (RED, GREEN, YELLOW)
- 10-pin JTAG connector for debug
- Two Grove expansion connectors (UART and I2C) for connectivity for additional sensors
- One PMOD expansion connector (SPI) for connectivity for additional peripherals



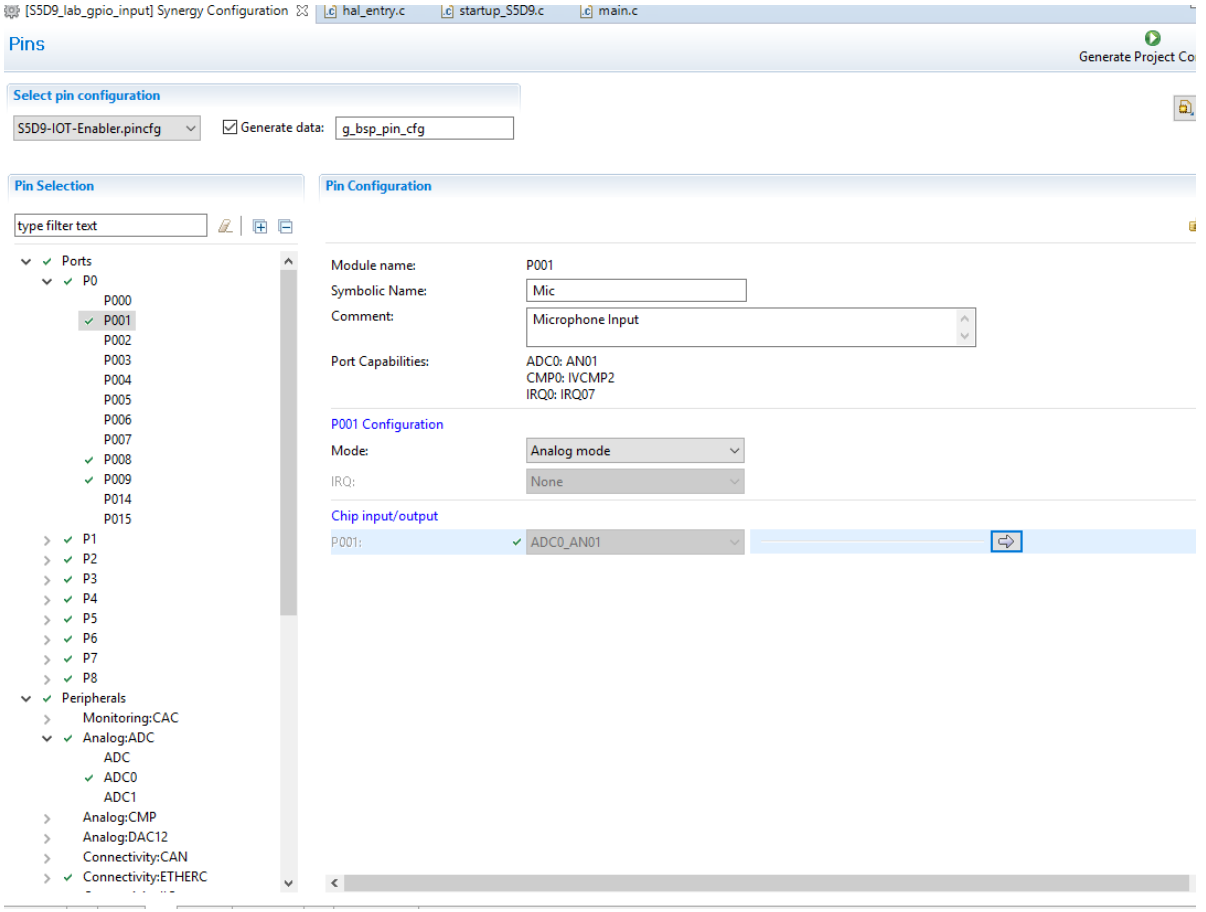


ADC MIC by Michael C. Li

ADC enabled

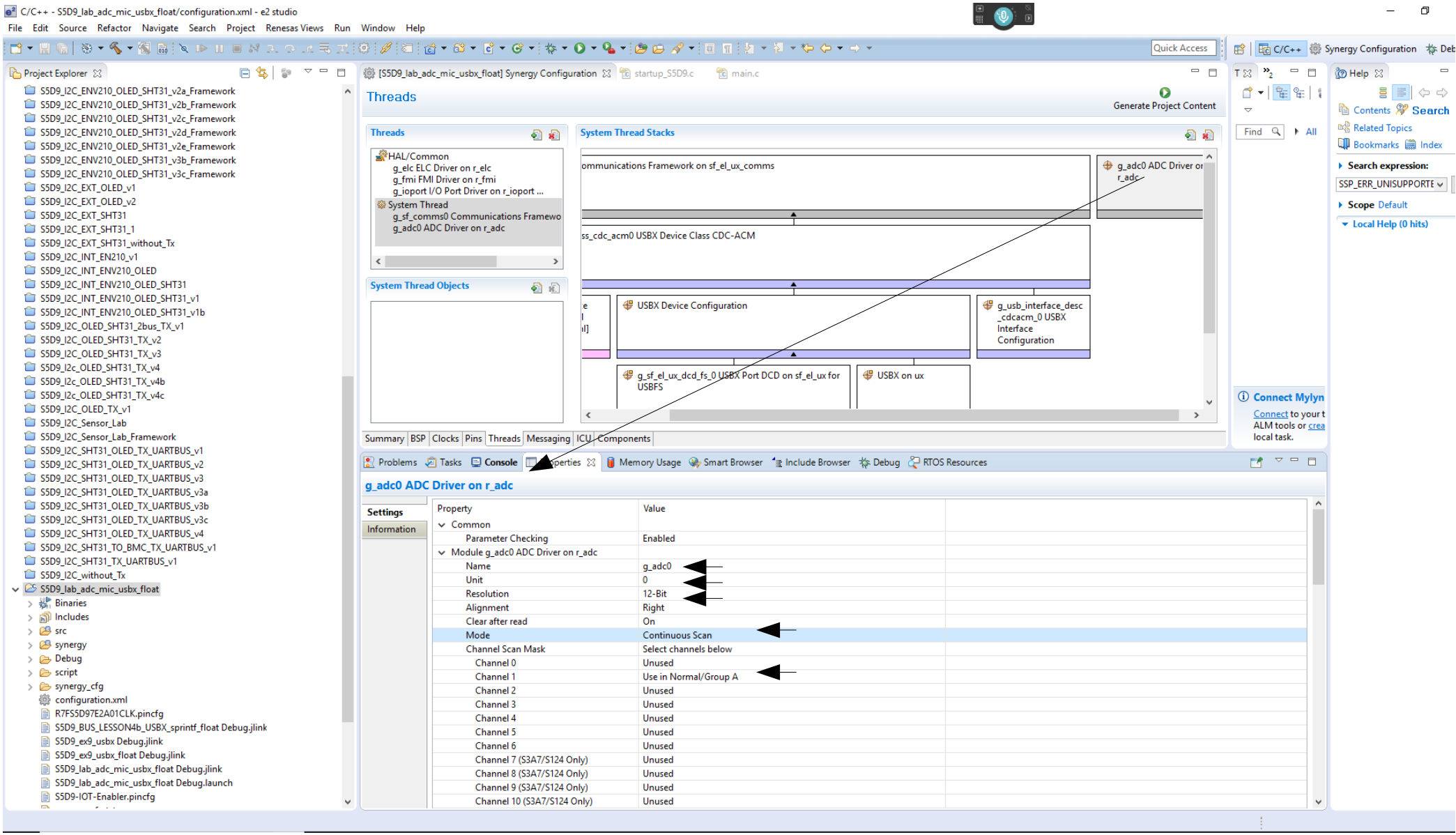

[S5D9_lab_gpio_input] Synergy Configuration
hal_entry.c
startup_S5D9.c
main.c
Pins
Generate Project Conte
Select pin configuration
S5D9-IOT-Enabler.pincfg
Generate data:
g_bsp_pin_cfg
Pin Selection
Pin Configuration
type filter text
Ports
P0
P1
P2
P3
P4
P5
P6
P7
P8
Peripherals
Monitoring:CAC
Analog:ADC
ADC
ADC0
ADC1
Analog:CMP
Analog:DAC12
Connectivity:CAN
Connectivity:ETHERC
Connectivity:IIC
Module name:
ADC0
Usage:
The below signal list may contain dedicated pins that cannot be configured and are shown for clarity
Operation Mode:
Custom
Input/Output
ADTRG:
None
AN00:
None
AN01:
P001
AN02:
None
AN03:
None
AN04:
None
AN05:
None
AN06:
None
AN07:
None

Pin Configuration

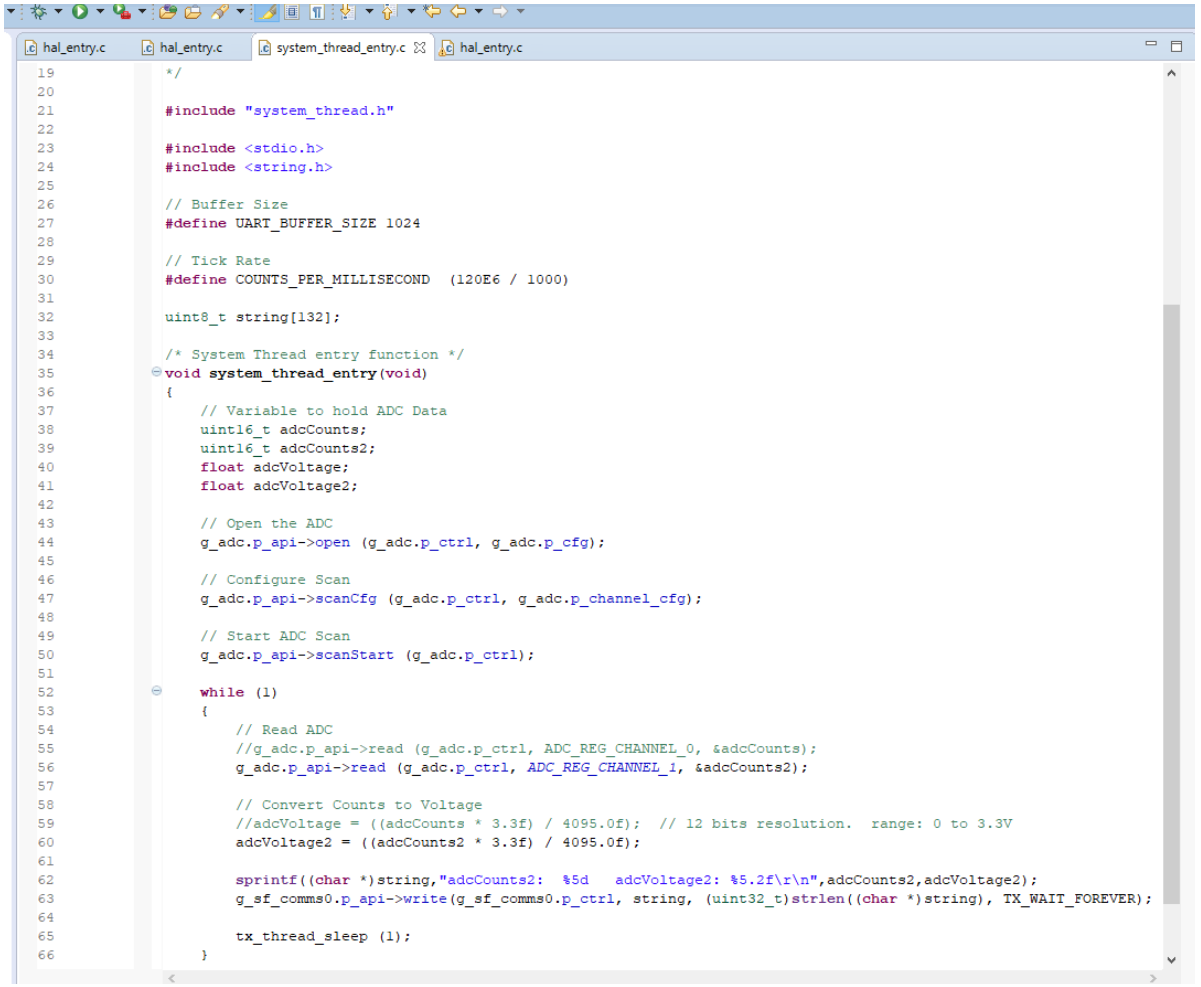


Set up ADC driver

(USBX is also set up to stream ADC data to PC)



Main Code




```
*/

#include "system_thread.h"

#include <stdio.h>
#include <string.h>

// Buffer Size
#define UART_BUFFER_SIZE 1024

// Tick Rate
#define COUNTS_PER_MILLISECOND  (120E6 / 1000)

uint8_t string[132];

/* System Thread entry function */
void system_thread_entry(void)
{
    // Variable to hold ADC Data
    uint16_t adcCounts;
    uint16_t adcCounts2;
    float adcVoltage;
    float adcVoltage2;

    // Open the ADC
    g_adc.p_api->open (g_adc.p_ctrl, g_adc.p_cfg);

    // Configure Scan
    g_adc.p_api->scanCfg (g_adc.p_ctrl, g_adc.p_channel_cfg);

    // Start ADC Scan
    g_adc.p_api->scanStart (g_adc.p_ctrl);

    while (1)
    {
        // Read ADC
        //g_adc.p_api->read (g_adc.p_ctrl, ADC_REG_CHANNEL_0, &adcCounts);
        g_adc.p_api->read (g_adc.p_ctrl, ADC_REG_CHANNEL_1, &adcCounts2);

        // Convert Counts to Voltage
        //adcVoltage = ((adcCounts * 3.3f) / 4095.0f);  // 12 bits resolution.  range: 0 to 3.3V
        adcVoltage2 = ((adcCounts2 * 3.3f) / 4095.0f);

        sprintf((char *)string,"adcCounts2:  %5d   adcVoltage2: %5.2f\r\n",adcCounts2,adcVoltage2);
        g_sf_comms0.p_api->write(g_sf_comms0.p_ctrl, string, (uint32_t)strlen((char *)string), TX_WAIT_FOREVER);

        tx_thread_sleep (1);
    }
```

Analog Voltage from MIC as I tap on it.