RE report#2 - Professor Kurian

Memo: NUPUR#2

Title: Data extraction from 1 PSoC-5 device

Author-Name: Nupur Patil

Author-Email: Nupur.Patil@iiitb.ac.in

Date: 18th February 2025

Summary

During the past week, the main objective was to extract data from a **PSoC-5 board** and store it in a **.csv file** for analysis. This involved configuring the pin layout to correctly interface with the ADC channels and setting up **UART communication** for data transmission. The board was programmed to sample analog signals and convert them into digital values, which were then transmitted over UART. A **Python script** was used to capture this data and save it in a .csv file in a structured format.

A total of **6000 readings** were collected from multiple ADC channels, ensuring accurate and consistent data sampling. The process included configuring the hardware, programming the device, and establishing reliable communication between the PSoC-5 board and the computer.

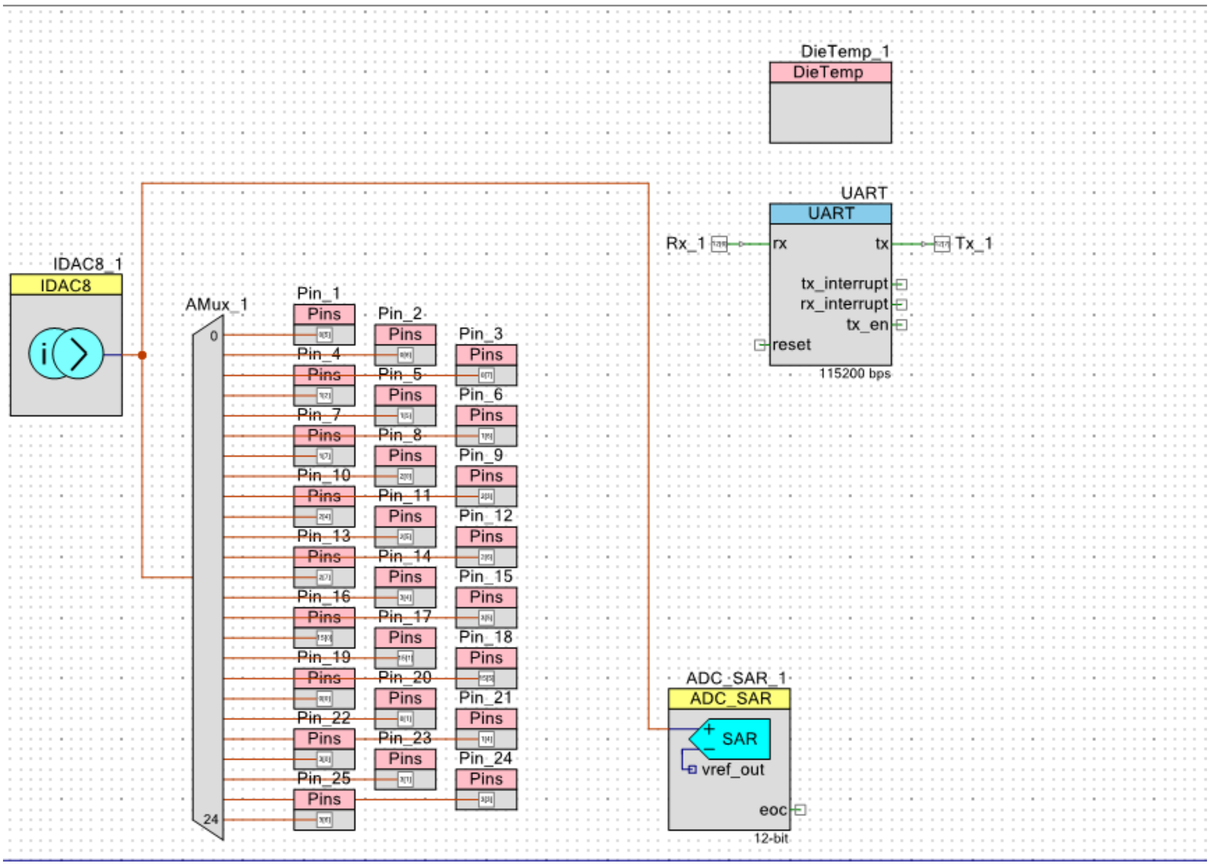
Details

The PSoC-5 board used for this task is the CY8C5888LTI-LP097 with a 68-QFN package.

The schematic design for the PSoC-5 board involves the following components:

- **AMux (Analog Multiplexer):** Used to switch between 25 different analog input channels for ADC conversion.
- **ADC (Analog-to-Digital Converter):** Converts the analog signals from the selected channel to digital values.
- **IDAC8 (Current Digital-to-Analog Converter):** Configured as a constant current source to provide a stable reference or bias current to the sensors. The value is set in the firmware to provide a fixed bias current.
- **UART (Universal Asynchronous Receiver/Transmitter):** Handles serial communication for transmitting the collected data to the host PC.
- **DieTemp Component:** measures the internal temperature of the PSoC-5 chip in degrees celsius.

Schematic Design



1. AMux (Analog Multiplexer)

- **Purpose:** The AMux component is used to select one out of 24 analog input channels at a time, which are then fed into the ADC for conversion.
- **Configuration:**
    - **Number of Channels:** 25 channels were configured to accommodate multiple analog inputs.
    - **Channel Switching:** The firmware sequentially switches between these channels to read the corresponding analog signals.
    - **Connection:** The output of the AMux is connected to the input of the ADC.

2. ADC (Analog-to-Digital Converter)

- **Purpose:** The ADC converts the analog signals received from the selected AMux channel into a 16-bit digital value.
- **Configuration:**
    - **Sampling Rate:** Controlled in the firmware to ensure accurate and stable readings.
    - **Triggering:** The ADC is triggered in the firmware after channel selection and a brief delay to allow signal stabilization.
- **Working:**
    - After the AMux selects a channel, the ADC is started, and conversion is performed.

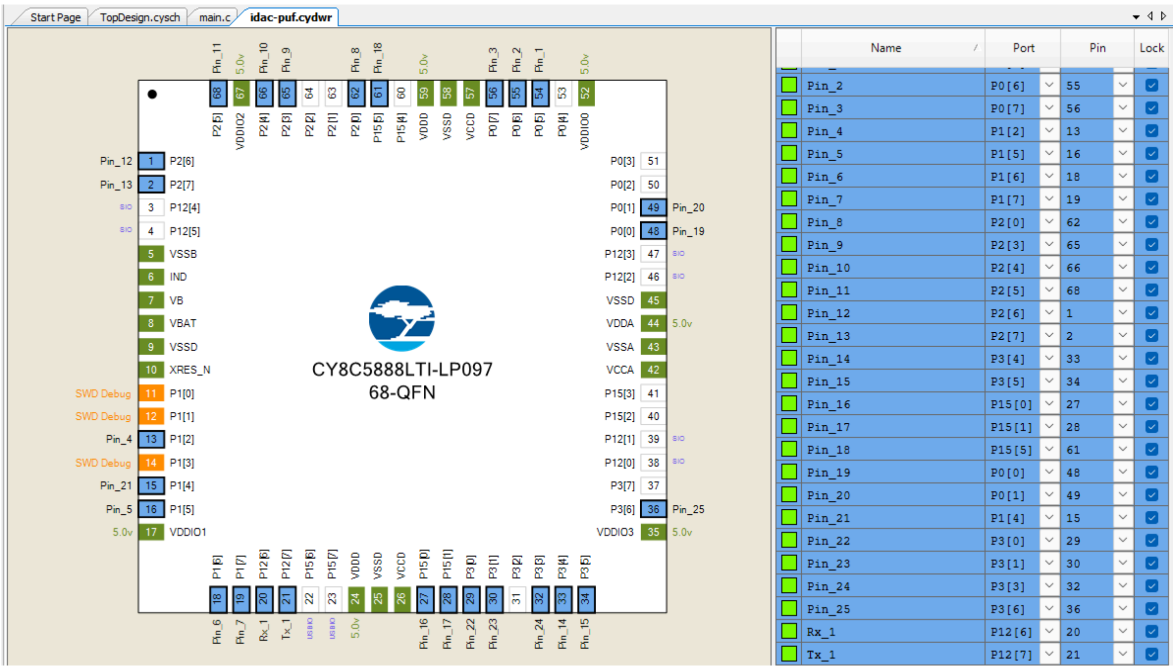
  - Once the conversion is complete, the digital value is stored in an array.

3. UART (Universal Asynchronous Receiver/Transmitter)

- **Purpose:** UART is used for serial communication between the PSoC-5 and the host PC.
- **Configuration:**
  - **Baud Rate:** Set to 115200 bps
  - **Mode:** Configured for asynchronous communication without flow control.
  - **Data Format:** 8 data bits, no parity, and 1 stop bit.(The default settings for UART in PSoC and for pySerial)
- **Working:**
  - The UART transmits the collected ADC data, the unique silicon ID, and the die temperature sequentially.
  - The firmware ensures data integrity by sending start and stop markers.

Control Logic and Flow

- **Sequence:**
  - The firmware initializes all components and enables global interrupts.
  - The AMux selects the first channel, and a delay is provided for signal stabilization.
  - The ADC converts the analog input to a digital value, which is stored in an array.
  - Steps 2 and 3 are repeated for all 25 channels.
  - The collected data, along with the unique silicon ID and die temperature, is transmitted over UART.
- **Start and Stop Markers:**
  - The UART transmission starts with `0x0D` and `0x0A` (Carriage Return and Line Feed).
  - It ends with `0x00`, `0xFF`, and `0xFF` as stop markers.

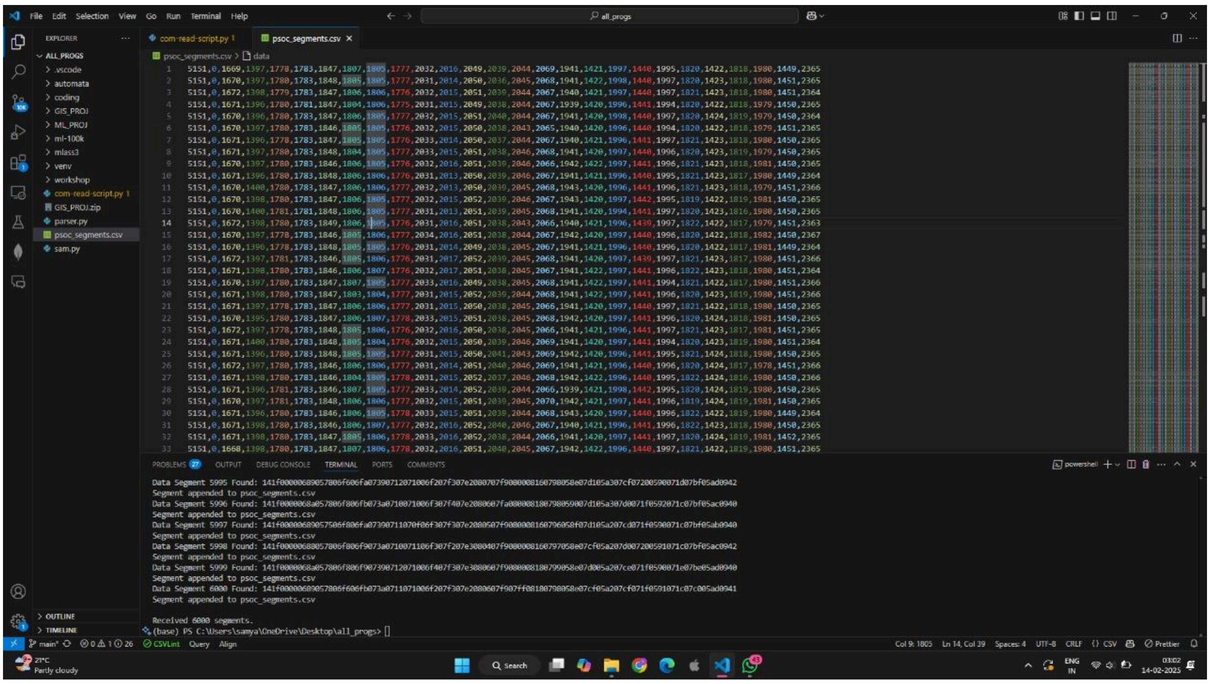


Firmware Implementation - [main.c]

- Initialization of components (ADC, AMux, UART, IDAC8).
- Sequentially switching between 25 channels using the AMux.
- Reading ADC values and storing them in an array.
- Transmitting ADC results, unique silicon ID, and die temperature over UART.

Data Collection - [python_script and the .csv generated]

Data was collected using a Python script that read the UART transmission and saved it to a .csv file.



psoc_segments

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	5151	0	1669	1397	1778	1783	1847	1807	1805	1777	2032	2016	2049	2039	2044	2069	1941	1421	1997	1440	1995	1820	1422	1818	1980	1449	2365
2	5151	0	1670	1397	1780	1783	1848	1805	1805	1777	2031	2014	2050	2036	2045	2068	1941	1422	1998	1440	1997	1820	1423	1818	1980	1451	2365
3	5151	0	1672	1398	1779	1783	1847	1806	1806	1776	2032	2015	2051	2039	2044	2067	1940	1421	1997	1440	1997	1821	1423	1818	1980	1451	2364
4	5151	0	1671	1396	1780	1781	1847	1804	1806	1775	2031	2015	2049	2038	2044	2067	1939	1420	1996	1441	1994	1820	1422	1818	1979	1450	2365
5	5151	0	1670	1396	1780	1783	1847	1806	1805	1777	2032	2015	2051	2040	2044	2067	1941	1420	1998	1440	1997	1820	1424	1819	1979	1450	2364

Understanding the Format

The data is structured as follows:

1. **Timestamp or Sequence Number:** The first value(5151) is the identifier(unique PSOC device) for this set of readings.
2. **Temperature:** here 0 is the reading
3. **Channel Numbers:** The subsequent 25 values correspond to the ADC values for channels 1 to 25 in sequential order.
4. **ADC Values:**
   - Each value is a 16-bit integer representing the digital output from the ADC for the corresponding channel.
   - For example:
     - Channel 1: 1669
     - Channel 2: 1397
     - ...
     - Channel 25: 2365

Conclusion

The data extraction from the PSoC-5 device was successfully completed, with 6000 readings recorded and saved in a .csv file. Next week I am supposed to collect data from 25 such devices along with understanding the temperature and voltage variation.

END