PBio Research Report Fall 2020

Student: Patrick LaChapelle
Faculty Advisor: Steve Perlmutter
UW net ID: pgl3
Student ID no.: 1873062
Submitted: December 21, 2020

Abstract

The objective of this Primate Biology Undergraduate research project has been to establish a software interface between the TIVA C family of microcontrollers and the high-speed AD7616 Analog Digital Converter from Analog devices. In this report, I will begin by explaining the relevant components of this project and concisely detailing the problem to be solved. I will then outline the solution to the problem, as well as its implementation, in detail. Finally, I will give some simple examples of the solution output (in the form of UART screenshots) and detail the next steps to take with this project. Code for this project can be found at: [kelray/AD7616-High-Speed-Interface (github.com)](https://github.com/kelray/AD7616-High-Speed-Interface)

Overview

The goal of this project is to establish a Tiva C driver for interface with the AD7616 high-speed Analog-Digital Converter chip to prototype faster reading of neuron signals in primate brains. This quarter, I focused on using the AD7616 chip in parallel hardware mode to emphasize speed and simplicity. After initially establishing a connection with a single channel (GitHub code: ad7616_initial-interface folder inside the “TivaC” folder) in hardware mode, I then used the Burst sequencer to interface all 16 channels with each read (GitHub code: ROUTINE_AD7616 folder inside “TivaC” folder). Since the purpose of this project is to simply establish a TIVA language driver, any TIVA C device will suffice, since the code can later be applied to any other device in the family. At $20 and with 40 GPIO ports, the TM4C1294XL launchpad boasts the low price point, ease-of-use, and flexibility desirable for prototyping projects, while still allowing for the large input/output required for a parallel interface in hardware mode. Furthermore, this TIVA C launchpad has a high clock rate of 120MHz, which further promotes the objective of speed.

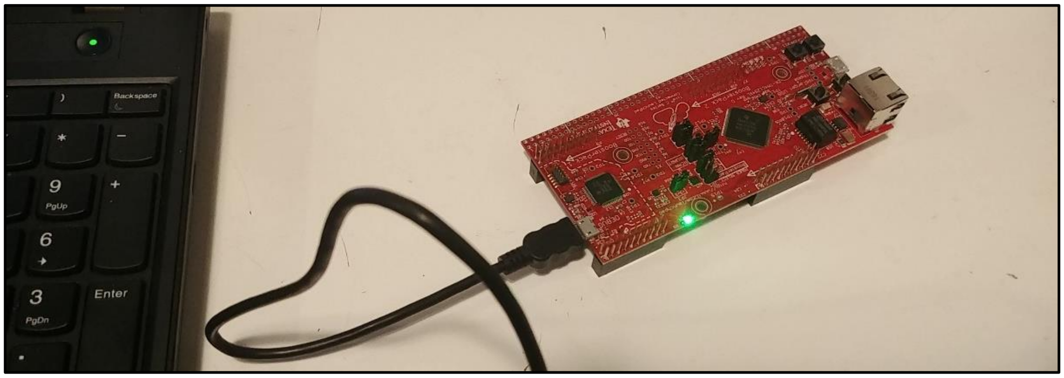


Fig. 1 - The Tiva C TM4C1294XL launchpad used for this project.

Like all other components in this project, the AD7616 PCB board for this was provided by Karim, who designed it for ease-of-use with firmware projects. This allowed me to focus mostly on the coding aspect, as opposed to solving inconsistencies from hardware issues. The AD7616 takes several logic inputs when operating in hardware mode: CONVST, RESET, CS, SEQEN, SER/PAR & BURST/WR, and outputs an important logic value, BUSY, along with the output data. Some of these are static for our purposes, and can be tied to either ground or high, while others are changing and need to be connected to GPIO pins for dynamic input/output. The logic inputs/outputs for the AD7616 use +3.3V logic levels, and the board requires an additional +5V power supply for operation. There are two groups of channels, A & B, which each have 8 channels for a total of 16. Each of these 16 channels has a grounded reference and a voltage to be read. To facilitate so many connections with some sort of order and clarity (there are a total of 53 connections) I add two breadboards and arrange the multicolored wired in alphabetical order according to the table of connections (the table of connections is discussed later, and can be found in the appendix).

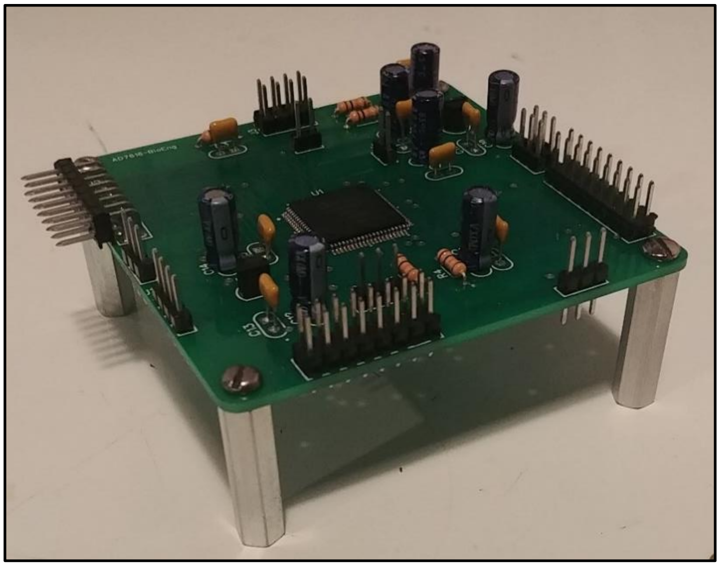


Fig. 2 The AD7616 development board

Hardware Implementation

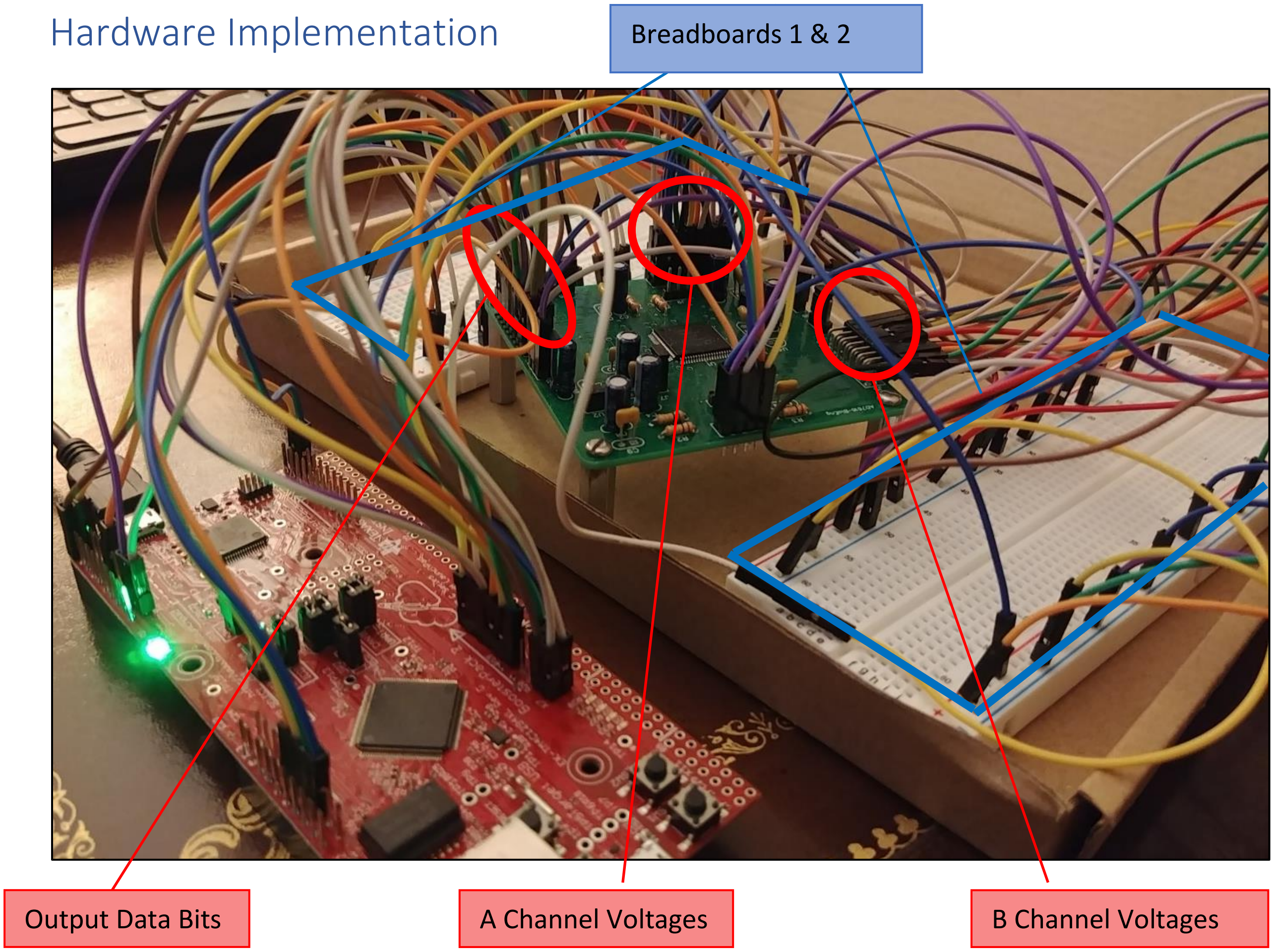


Fig. 3 The complete setup for the Ad7616 – TM4C1294XL interface. In addition to the two development boards, I added two spare breadboards I had to make the connections easier.

Above in Figure 3, one can see the setup for this interface, and I have labeled the breadboards along with the A & B channel inputs and the output data bits (DB0-DB15). The 16 output data bits connect to the K & M GPIO ports on the Tiva C launchpad. The TM4C1294XL has to voltage output levels: 5V & 3.3V, which are connected to rails on the breadboard along with the ground pin, then to the AD7616 accordingly.

Logical implementation of operation

The logical implementation of this setup begins with a complete reset of the AD7616, which is initiated by manipulating the PP3 GPIO pin, which is configured as +3.3V logic output and connected to the RESET pin on the AD7616, according to the reset specifications in the AD7616 documentation. Furthermore, the other hardware inputs are configured according to the initial setup specifications in the documentation, and the chip is ready to begin converting voltages. Currently, the code implements a single timer, which waits for designated times specified in the AD7616 documentation between operations. This is somewhat inefficient, and future implementations of this routine will need to operate solely using interrupt routines to increase speed and facilitate scalability and applicability. As previously mentioned, the AD7616 is configured to operate in parallel hardware mode, the timing diagram for this configuration is included below.

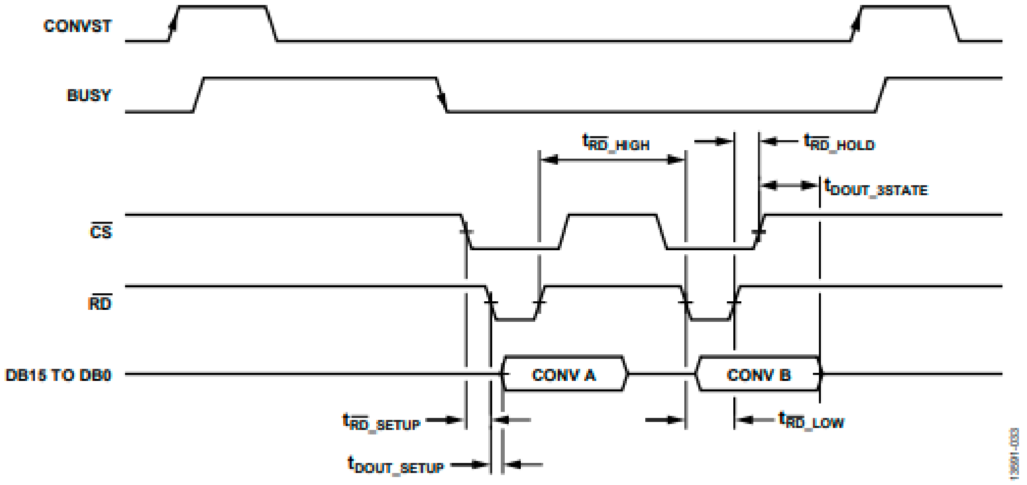


Figure 4. Parallel Read Timing Diagram

Fig. 4 – The timing diagram for the AD7616 in hardware mode.

Additionally, the chip is configured in burst sequencer mode by setting the BURST and SEQEN pins to high, so the code will need to run through the reads for the A & B channel conversions 8 times per each CONVST signal, since burst sequencer mode converts all specified channels with a single CONVST pulse. The logic is as follows below:

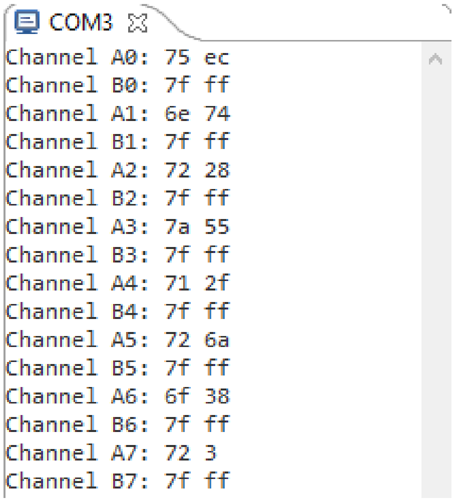
```
While the program runs:
    Initiate conversion by tying CONVST high
    Wait 50 ns to ensure the CONVST signal registers
    Tie CONVST low to ensure a new conversion does not begin
    Wait max. time of 4.2 µs for BUSY to go low
    Begin Read loop:
        Wait 20ns to setup CS
        Tie CS low to begin a read operation
        Wait 10ns to setup RD
        Tie RD low to continue read operation setup
        Wait 30ns for the data to appear on the AD7616 output data bits
        Send RD high
        Read the channel data to variables in TIVA C memory
        Send CS high
        Repeat Read loop 16 times for channels A0-B7
```

Print New data to the UART

Return to the beginning of the program loop to initiate a new conversion.

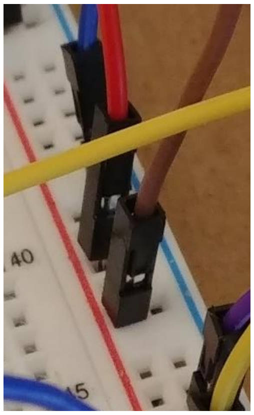
Program Output

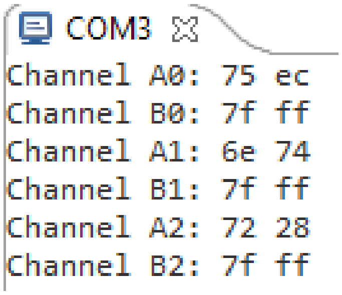
The program reads the voltage conversion values to the UART. Figure 5 shows the complete output, while figure 6 shows some of this output and demonstrates how it is associated with the voltage values. The voltage values output from the AD7616 are 16 binary bits, which can be represented by 4 hex bits. The voltage is given in two's complement format. This means a reading of 0111 (7 in hex format) on the four most significant bits indicates a near-maximum value, while a reading of 1111 (f in hex format) indicates a near-minimum value. As exhibited in figure 5.A, all B channels are showing absolute maximum values, while most A channels show near-maximum values. The initial "test" setup for this project uses all voltages reading values of +3.3V, and the reference voltage for the AD7616 set to +2.5V, so that we should be reading maximum values. We have concluded that this is likely due to the noisy nature of the setup; several cheap wires, 2 breadboards and the AD7616 PCB leaves this setup with multiple sources of noise. Figure 5.B-5.E demonstrate how the UART is showing the voltage values, and its ability to update in real-time.



```
Channel A0: 75 ec
Channel B0: 7f ff
Channel A1: 6e 74
Channel B1: 7f ff
Channel A2: 72 28
Channel B2: 7f ff
Channel A3: 7a 55
Channel B3: 7f ff
Channel A4: 71 2f
Channel B4: 7f ff
Channel A5: 72 6a
Channel B5: 7f ff
Channel A6: 6f 38
Channel B6: 7f ff
Channel A7: 72 3
Channel B7: 7f ff
```

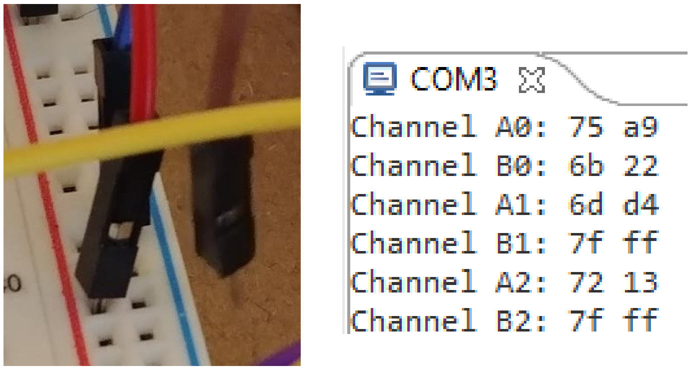
Fig. 5 – The UART output display for this initial interface routine. The noted discrepancies between A and B channels are exhibited here.





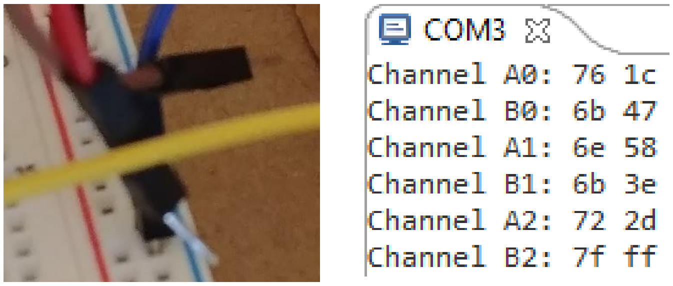
```
Channel A0: 75 ec
Channel B0: 7f ff
Channel A1: 6e 74
Channel B1: 7f ff
Channel A2: 72 28
Channel B2: 7f ff
```

Fig. 6.A – Channels B0 & B1 plugged into the high +3.3V, along with the UART output for this setup



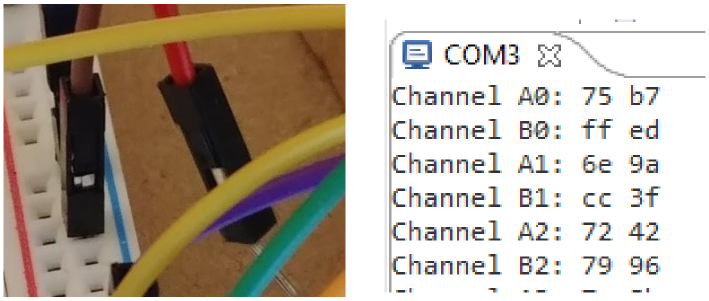
```
Channel A0: 75 a9
Channel B0: 6b 22
Channel A1: 6d d4
Channel B1: 7f ff
Channel A2: 72 13
Channel B2: 7f ff
```

Fig. 6.B – Channel B1 plugged into the high +3.3V, and B0 left floating along with the UART output for this setup



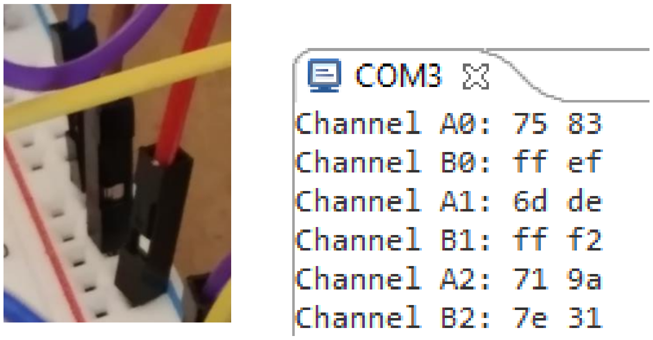
```
Channel A0: 76 1c
Channel B0: 6b 47
Channel A1: 6e 58
Channel B1: 6b 3e
Channel A2: 72 2d
Channel B2: 7f ff
```

Fig. 6.C – Channels B0 & B1 left floating, along with the UART output for this setup



```
Channel A0: 75 b7
Channel B0: ff ed
Channel A1: 6e 9a
Channel B1: cc 3f
Channel A2: 72 42
Channel B2: 79 96
```

Fig. 6.D – Channel B0 connected to ground and channel B1 left floating, along with the UART output for this setup



```
Channel A0: 75 83
Channel B0: ff ef
Channel A1: 6d de
Channel B1: ff f2
Channel A2: 71 9a
Channel B2: 7e 31
```

Fig. 6.E – Channels B0 & B1 plugged into gnd, along with the UART output for this setup

Discussion and Future Plans

As mentioned previously, rewriting this code so that it uses hardware interrupts (as opposed to the single timer interrupt currently being utilized) is desirable to promote speed, efficiency, and scalability. Future work includes implementing these changes, as well as further adding complexity to the interface. Functions need to be implemented for single channel viewing, user manipulated channel selection, and for data storage. Finally, thorough analysis needs to be conducted for the interface and the setup should be evaluated on a basis of accuracy, precision, reliability, speed and efficiency.

Appendix

Table A1 – The connections for the initial interface, found in the AD7616_initial-interface folder

AD7616 REGISTER	AD7616 EVAL Connector	Pin no.	Connection on breadboard/TM4C1294XL
DB0	J2	1	GPIO PK0 (INPUT)
DB1	J2	2	GPIO PK1 (INPUT)
DB2	J2	3	GPIO PK2 (INPUT)
DB3	J2	4	GPIO PK3 (INPUT)
DB4	J2	5	GPIO PK4 (INPUT)
DB5	J2	6	GPIO PK5 (INPUT)
DB6	J2	7	GPIO PK6 (INPUT)
DB7	J2	8	GPIO PK7 (INPUT)
DB8	J2	9	GPIO PM0 (INPUT)
DB9	J2	10	GPIO PM1 (INPUT)
DB10	J2	11	GPIO PM2 (INPUT)
DB11	J2	12	GPIO PM3 (INPUT)
DB12	J2	13	GPIO PM4 (INPUT)
DB13	J2	14	GPIO PM5 (INPUT)
DB14	J2	15	GPIO PM6 (INPUT)
DB15	J2	16	GPIO PM7 (INPUT)
CONVST	J3	1	GPIO PP5 (OUTPUT, CONVST)
BUSY	J3	2	GPIO PA7 (INPUT)
Reset input	J3	3	GPIO PP3 (OUTPUT)
RD (Parallel data read)	J3	4	GPIO PQ1 (OUTPUT)
CS	J3	5	GPIO PC6 (OUTPUT)
SEQEN	J3	6	GND (Breadboard)
SER/PAR	J3	7	GND (Breadboard)
BURST/WR	J3	8	GND (Breadboard)
V0A	J4	1	+3.3V (Breadboard)
V0AGND	J4	2	GND (Breadboard)
V1A	J4	3	+3.3V (Breadboard)
V1AGND	J4	4	GND (Breadboard)
V2A	J4	5	+3.3V (Breadboard)
V2AGND	J4	6	GND (Breadboard)
V3A	J4	7	+3.3V (Breadboard)
V3AGND	J4	8	GND (Breadboard)
V4A	J4	9	+3.3V (Breadboard)
V4AGND	J4	10	GND (Breadboard)
V5A	J4	11	+3.3V (Breadboard)
V5AGND	J4	12	GND (Breadboard)
V6A	J4	13	+3.3V (Breadboard)
V6AGND	J4	14	GND (Breadboard)
V7A	J4	15	+3.3V (Breadboard)
V7AGND	J4	16	GND (Breadboard)
V0B	J5	1	+3.3V (Breadboard)

V0BGND	J5	2	GND (Breadboard)
V1B	J5	3	+3.3V (Breadboard)
V1BGND	J5	4	GND (Breadboard)
V2B	J5	5	+3.3V (Breadboard)
V2BGND	J5	6	GND (Breadboard)
V3B	J5	7	+3.3V (Breadboard)
V3BGND	J5	8	GND (Breadboard)
V4B	J5	9	+3.3V (Breadboard)
V4BGND	J5	10	GND (Breadboard)
V5B	J5	11	+3.3V (Breadboard)
V5BGND	J5	12	GND (Breadboard)
V6B	J5	13	+3.3V (Breadboard)
V6BGND	J5	14	GND (Breadboard)
V7B	J5	15	+3.3V (Breadboard)
V7BGND	J5	16	GND (Breadboard)
CHSEL0	J8	1	GND (Breadboard)
CHSEL1	J8	2	GND (Breadboard)
CHSEL2	J8	3	GND (Breadboard)
HW_RNGSEL0	J8	4	GND (Breadboard)
HW_RNGSEL1	J8	5	+3.3V (Breadboard)
+5V input	J10	1	+5V (Breadboard)
+3.3V input	J10	4	+3.3V (Breadboard)

Table A2 – The connections for the second interface setup

AD7616 REGISTER	AD7616 EVAL Connector	Pin no.	Connection on breadboard/TM4C1294XL
DB0	J2	1	GPIO PK0 (INPUT)
DB1	J2	2	GPIO PK1 (INPUT)
DB2	J2	3	GPIO PK2 (INPUT)
DB3	J2	4	GPIO PK3 (INPUT)
DB4	J2	5	GPIO PK4 (INPUT)
DB5	J2	6	GPIO PK5 (INPUT)
DB6	J2	7	GPIO PK6 (INPUT)
DB7	J2	8	GPIO PK7 (INPUT)
DB8	J2	9	GPIO PM0 (INPUT)
DB9	J2	10	GPIO PM1 (INPUT)
DB10	J2	11	GPIO PM2 (INPUT)
DB11	J2	12	GPIO PM3 (INPUT)
DB12	J2	13	GPIO PM4 (INPUT)
DB13	J2	14	GPIO PM5 (INPUT)
DB14	J2	15	GPIO PM6 (INPUT)
DB15	J2	16	GPIO PM7 (INPUT)
CONVST	J3	1	GPIO PP5 (OUTPUT, CONVST)
BUSY	J3	2	GPIO PA7 (INPUT)
Reset input	J3	3	GPIO PP3 (OUTPUT)

RD (Parallel data read)	J3	4	GPIO PQ1 (OUTPUT)
CS	J3	5	GPIO PC6 (OUTPUT)
SEQEN	J3	6	+3.3V (Breadboard)
SER/PAR	J3	7	GND (Breadboard)
BURST/WR	J3	8	+3.3V (Breadboard)
V0A	J4	1	+3.3V (Breadboard)
V0AGND	J4	2	GND (Breadboard)
V1A	J4	3	+3.3V (Breadboard)
V1AGND	J4	4	GND (Breadboard)
V2A	J4	5	+3.3V (Breadboard)
V2AGND	J4	6	GND (Breadboard)
V3A	J4	7	+3.3V (Breadboard)
V3AGND	J4	8	GND (Breadboard)
V4A	J4	9	+3.3V (Breadboard)
V4AGND	J4	10	GND (Breadboard)
V5A	J4	11	+3.3V (Breadboard)
V5AGND	J4	12	GND (Breadboard)
V6A	J4	13	+3.3V (Breadboard)
V6AGND	J4	14	GND (Breadboard)
V7A	J4	15	+3.3V (Breadboard)
V7AGND	J4	16	GND (Breadboard)
V0B	J5	1	+3.3V (Breadboard)
V0BGND	J5	2	GND (Breadboard)
V1B	J5	3	+3.3V (Breadboard)
V1BGND	J5	4	GND (Breadboard)
V2B	J5	5	+3.3V (Breadboard)
V2BGND	J5	6	GND (Breadboard)
V3B	J5	7	+3.3V (Breadboard)
V3BGND	J5	8	GND (Breadboard)
V4B	J5	9	+3.3V (Breadboard)
V4BGND	J5	10	GND (Breadboard)
V5B	J5	11	+3.3V (Breadboard)
V5BGND	J5	12	GND (Breadboard)
V6B	J5	13	+3.3V (Breadboard)
V6BGND	J5	14	GND (Breadboard)
V7B	J5	15	+3.3V (Breadboard)
V7BGND	J5	16	GND (Breadboard)
CHSEL0	J8	1	+3.3V (Breadboard)
CHSEL1	J8	2	+3.3V (Breadboard)
CHSEL2	J8	3	+3.3V (Breadboard)
HW_RNGSEL0	J8	4	GND (Breadboard)
HW_RNGSEL1	J8	5	+3.3V (Breadboard)
+5V input	J10	1	+5V (Breadboard)
+3.3V input	J10	4	+3.3V (Breadboard)