Communication Protocols – UART and I^{2}C

Studio 7C – Week 7

Comp Eng 2DX3

McMaster University

Acknowledgments

- Successive versions of this studio were developed with the efforts of:
  - Thomas Doyle
  - Ama Simons
  - Hafez Mousavi
  - Yaser Haddara
  - Shahrukh Athar

McMaster University

This studio focuses on ***Communication Protocols***

We will cover:

1. UART/SCI – An asynchronous communications protocol
2. I^{2}C – A synchronous communications protocol

You will need

- Realterm – https://realterm.sourceforge.io/
- The MSP432E401Y
- The Analog Discovery 3 (AD3) and the WaveForms software
- Studio7C_Code.zip folder from Avenue
- Breadboard, wires, and resistors

Asynchronous vs. Synchronous Communication

- Asynchronous protocols (e.g. UART)
  - There is no clock signal to synchronize the output of bits
  - Communicate via an agreed baud rate in bps (bits per second)
  - No handshaking
- Synchronous protocols (e.g. I^2C)
  - Include a dedicated clock line that determines the rate of data transfer
  - Send one bit at each clock cycle
  - Leader/Follower handshaking

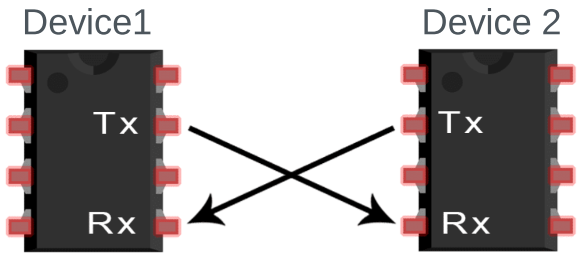
Asynchronous: UART / SCI

Universal Asynchronous Receiver-Transmitter (UART)

- A point-to-point Connection
- Two unidirectional Connection wires (Rx and Tx separately)

- Both devices have the same level of privilege (no leader/follower)

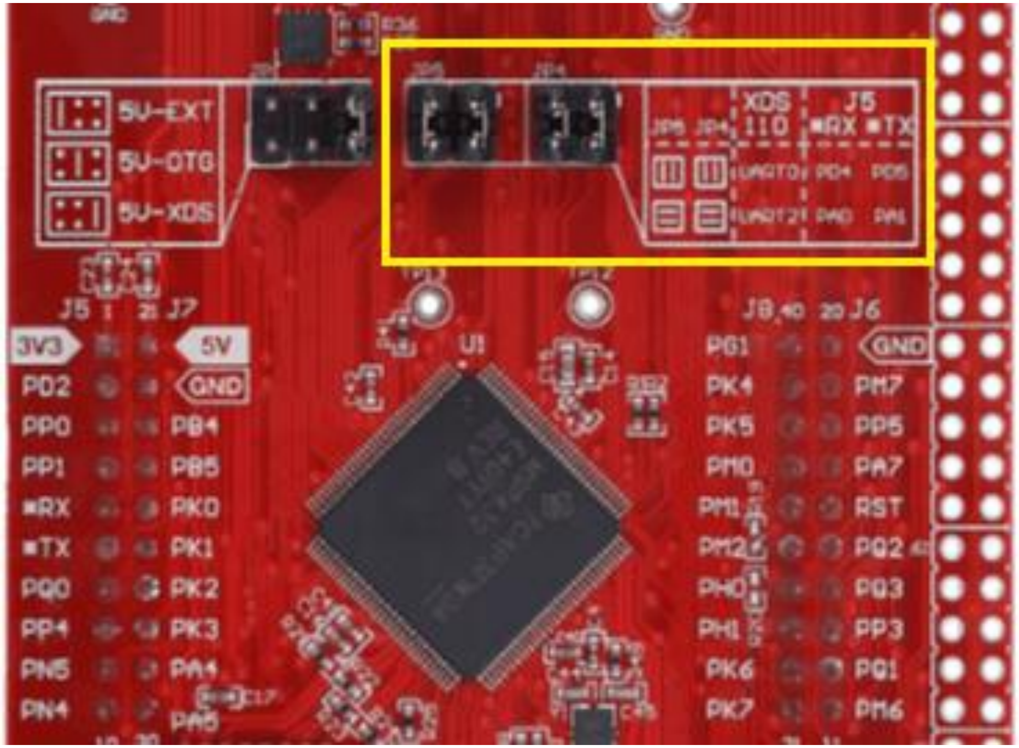
- No clocking is needed



- Although in your MCU, the UART is embedded in the USB port, you can think of it as simple wiring!

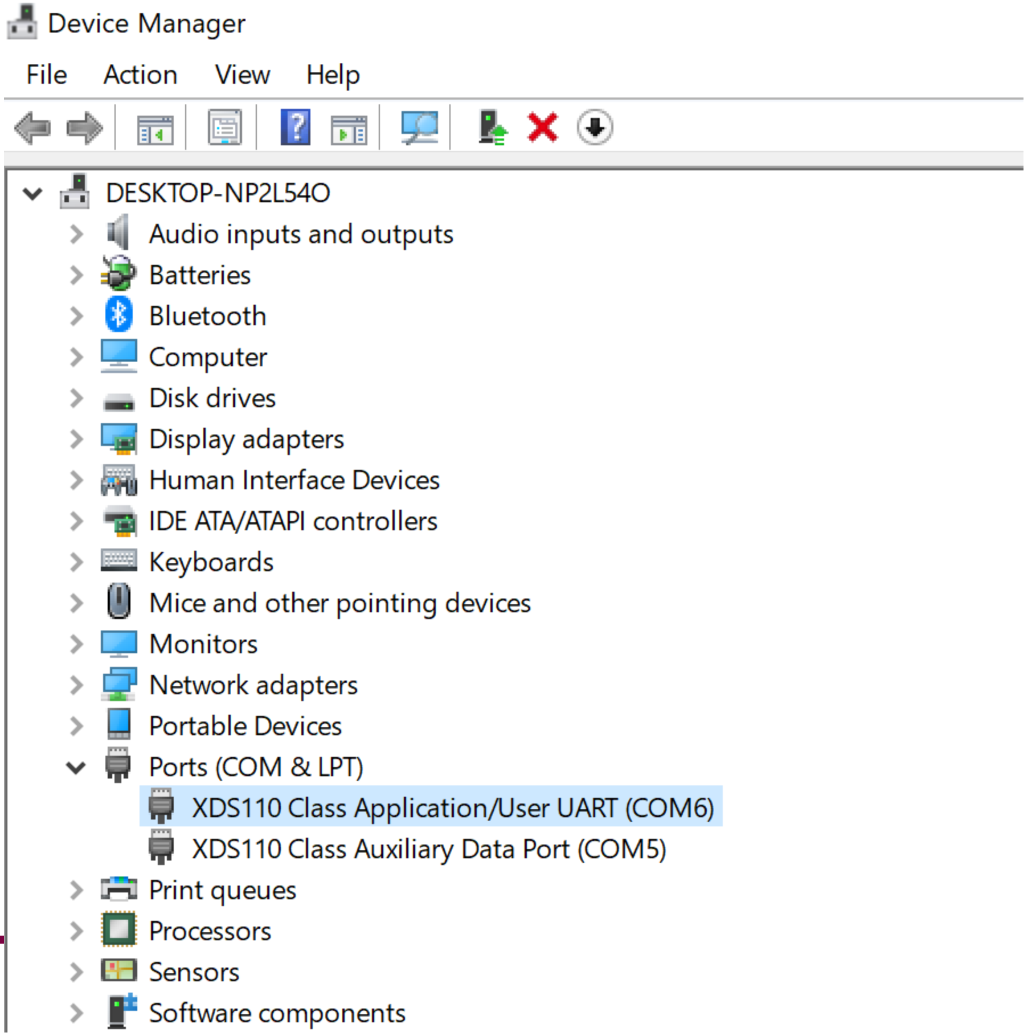
UART Hardware Setup

- Connect the micro to the computer
- Check the positions of jumpers JP4 & JP5 against this picture:
  - The vertical jumper position configures UART0 to connect through the USB port



Find the Port on PC

- Open Device Manager
- Look under Ports (COM & LPT)
- Find the XDS110 User UART
- Note your port number (6 in this example)


Device Manager

File Action View Help

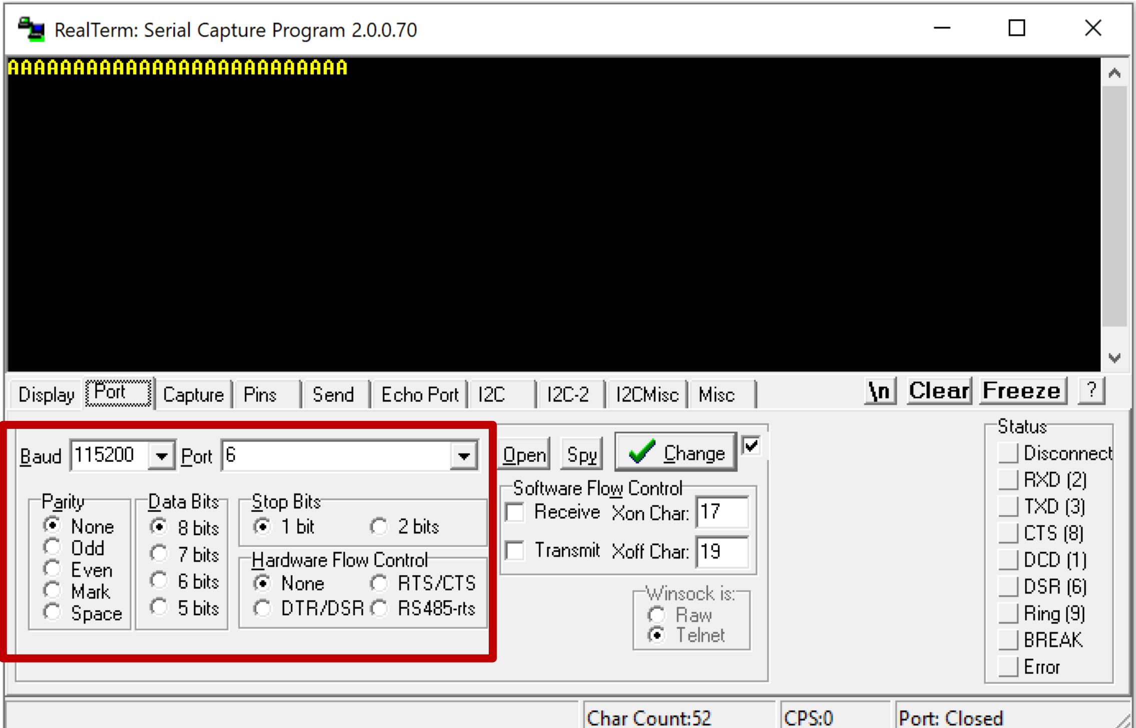
DESKTOP-NP2L54O
- Audio inputs and outputs
- Batteries
- Bluetooth
- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Portable Devices
- Ports (COM & LPT)
  - XDS110 Class Application/User UART (COM6)
  - XDS110 Class Auxiliary Data Port (COM5)
- Print queues
- Processors
- Sensors
- Software components

Using the sample code

- Download **Studio7C_Code.zip** and extract the files
- Open the **Studio7C_UART** project
- Read through the UART_Init() code
  - The comments tell you where each value is being configured
  - Baud rate: 115200 bps
  - No parity
  - 8 data bits
  - 1 stop bit
- Translate and Build the code and download to the MCU. Don’t start it yet.

Set the Configuration in RealTerm

- Download and install RealTerm (https://realterm.sourceforge.io/)
- Configure the Port Tab to match your MSP432E401Y UART settings:
  1. Make sure the correct port is selected (6 in this example)
  2. Baud rate: 115200
  3. Parity: None
  4. Data Bits: 8 bits
  5. Stop Bits: 1 bit
  6. Hardware Flow Control: None


RealTerm: Serial Capture Program 2.0.0.70

```
AAAAAAAAAAAAAAAAAAAAAAAAAA
```

Baud 115200 Port 6

Parity: [✓ None] Odd Even Mark Space

Data Bits: [✓ 8 bits] 7 bits 6 bits 5 bits

Stop Bits: [✓ 1 bit] 2 bits

Hardware Flow Control: [✓ None] RTS/CTS DTR/DSR RS485-rts

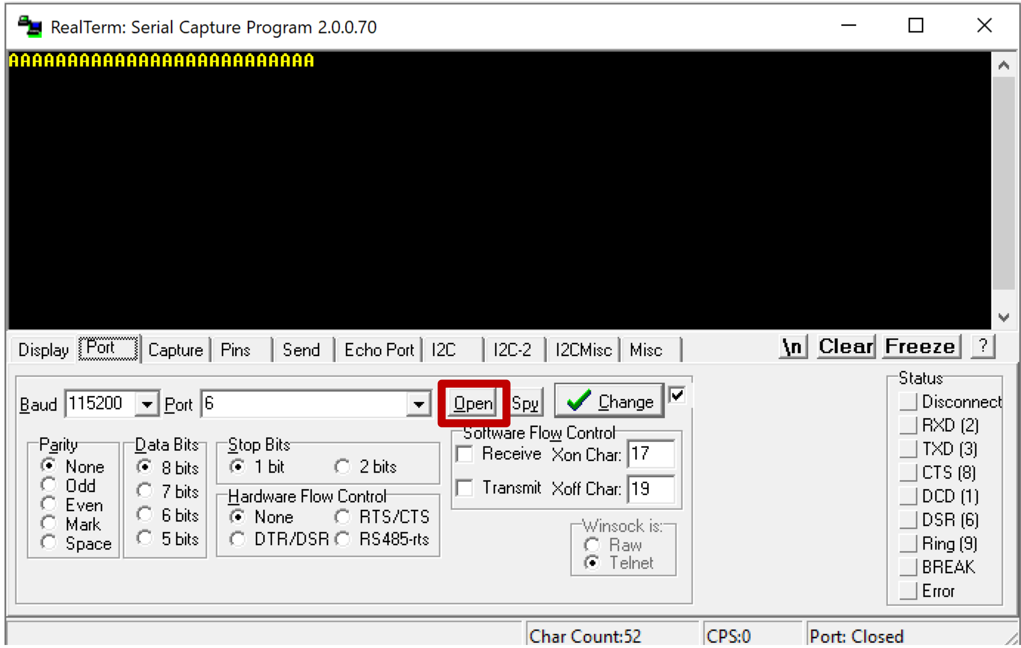
Software Flow Control: Receive Xon Char: 17 Transmit Xoff Char: 19

Winsock is: Raw [✓ Telnet]

Char Count:52 CPS:0 Port: Closed

Transmit Data to PC

- Click the "Open" button to open the port
- Reset the MCU to start transmission
- Watch the letter A being transmitted to RealTerm repeatedly
- Also note each time a transmission happens, onboard LED D2 blinks


RealTerm: Serial Capture Program 2.0.0.70

```
AAAAAAAAAAAAAAAAAAAAAAAAAA
```

Baud 115200 Port 6

Parity: [✓ None] Odd Even Mark Space

Data Bits: [✓ 8 bits] 7 bits 6 bits 5 bits

Stop Bits: [✓ 1 bit] 2 bits

Hardware Flow Control: [✓ None] RTS/CTS DTR/DSR RS485-rts

Software Flow Control: Receive Xon Char: 17 Transmit Xoff Char: 19

Winsock is: Raw [✓ Telnet]

Status: Disconnect RXD (2) TXD (3) CTS (8) DCD (1) DSR (6) Ring (9) BREAK Error

Char Count:52 CPS:0 Port: Closed

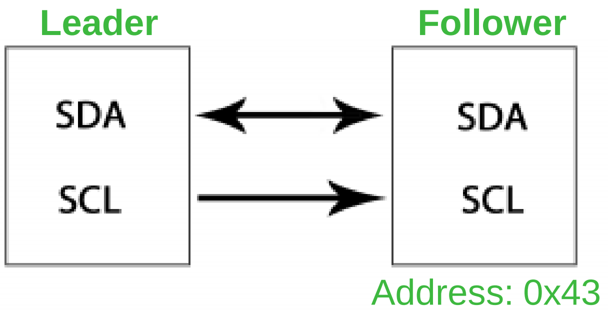
Modifying the code

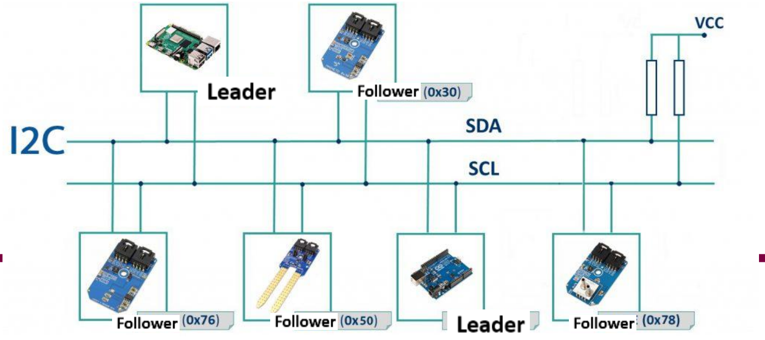
- Go through the code to find out the code segments responsible for the transmission of the letter ‘A’
- Change the code to transmit a different letter (let’s say the first letter of your first name) – check your result on RealTerm
- Change the code to transmit the string “Hello World!”
  - You will need to complete the code for the function **UART_OutString()**
  - You will need to comment out the **UART_OutChar()** function call and uncomment the **UART_OutString()** function call

Synchronous: I^{2}C

Synchronous Inter-Integrated Circuit (I^2C Protocol)

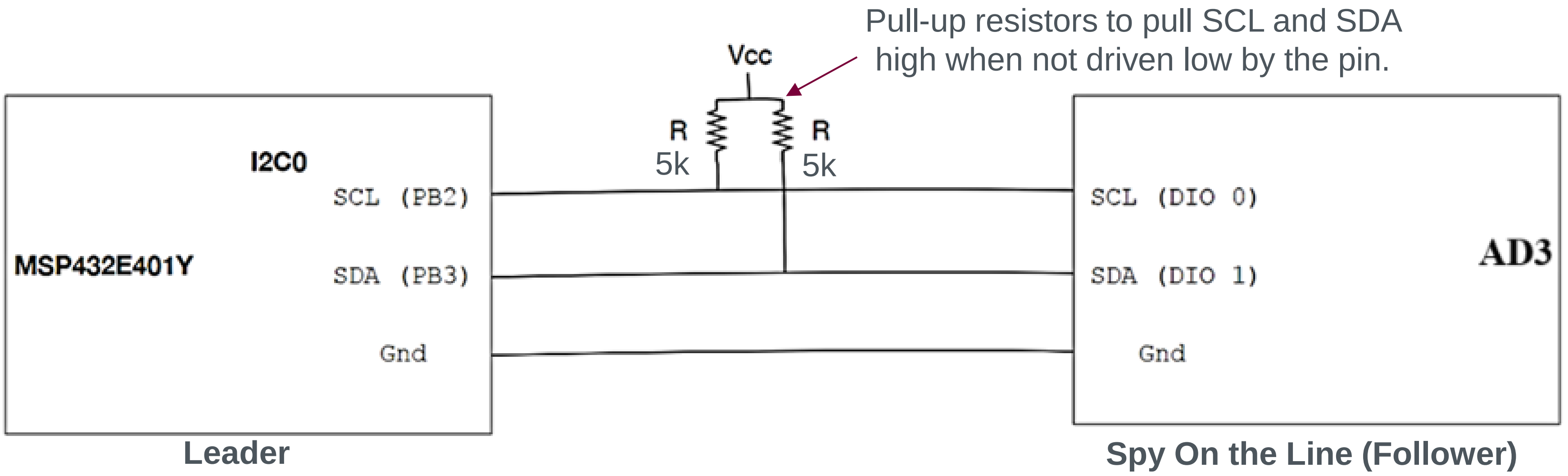
- **A Broadcast Network protocol. Many nodes can be connected to same bus.**
- There is a **Leader/Follower** relationship when communicating with I^2C
  - The Leader outputs the clock (determines the speed of data transfer)
- Leader can have many Followers, each recognized by a **unique address.**
- **I^2C interface uses two signal wires to exchange information**
  1. **Serial Data Line** (SDA) line for sending and receiving data
  2. **Serial Clock Line** (SCL) the clock signal line, synchronizes data transfer
- Note that in I^2C we have just one bidirectional RX/TX data line





Circuit Setup

- Connect the wires between the MSP432E401Y and AD3 as shown. This configuration uses I2C0 on your microcontroller. Use V_{cc} = 3.3 V
- The AD3 is acting as a spy on the line. It can listen to what MCU is writing on the bus.


Pull-up resistors to pull SCL and SDA high when not driven low by the pin.
Vcc
R
5k
R
5k
I2C0
SCL (PB2)
MSP432E401Y
SDA (PB3)
Gnd
Leader
SCL (DIO 0)
SDA (DIO 1)
AD3
Gnd
Spy On the Line (Follower)

Calculation of R: https://www.youtube.com/watch?v=sGZe0aJsqBQ

Circuit Setup

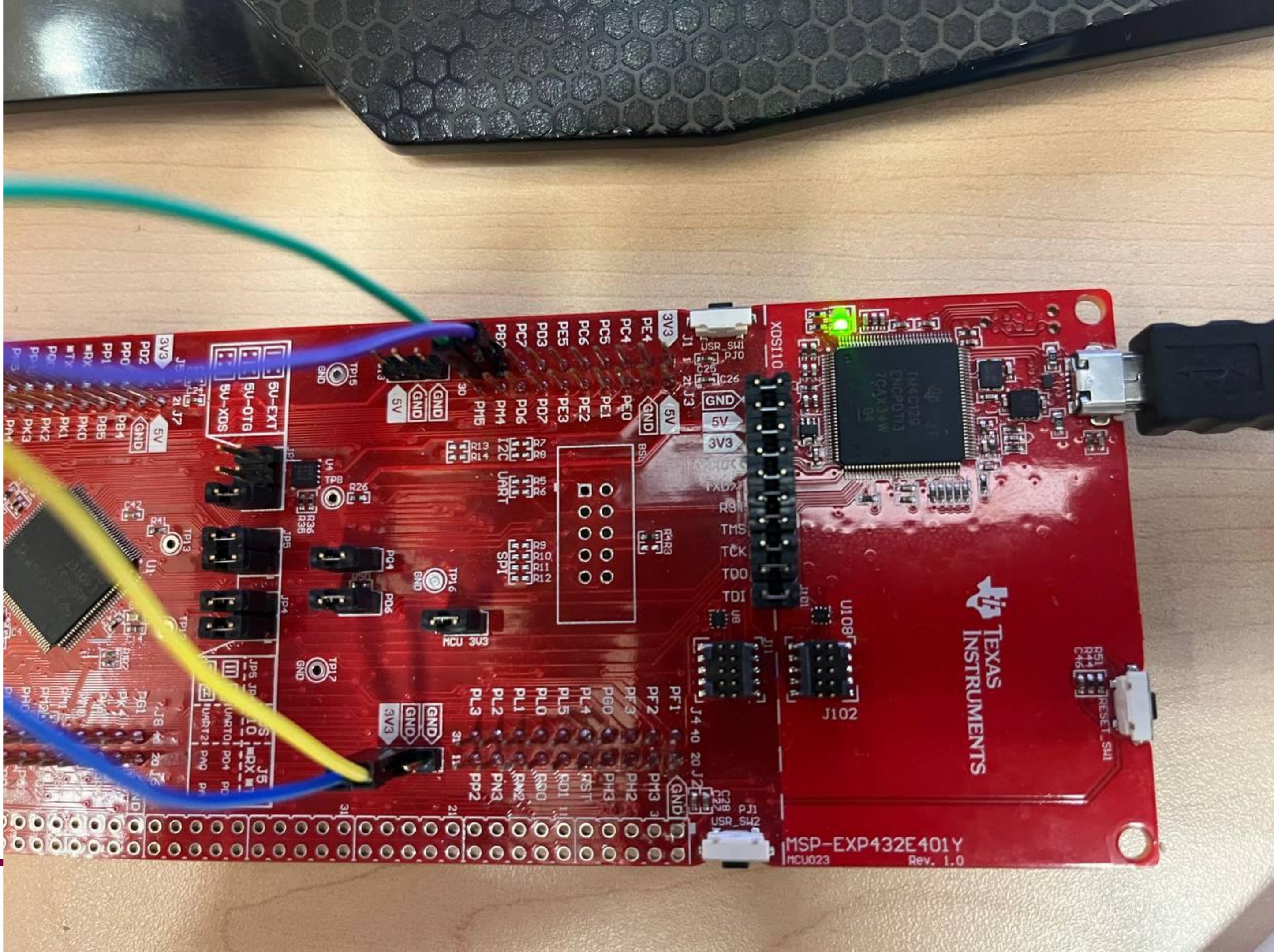
MCU side

Green (PB3)

Purple (PB2)

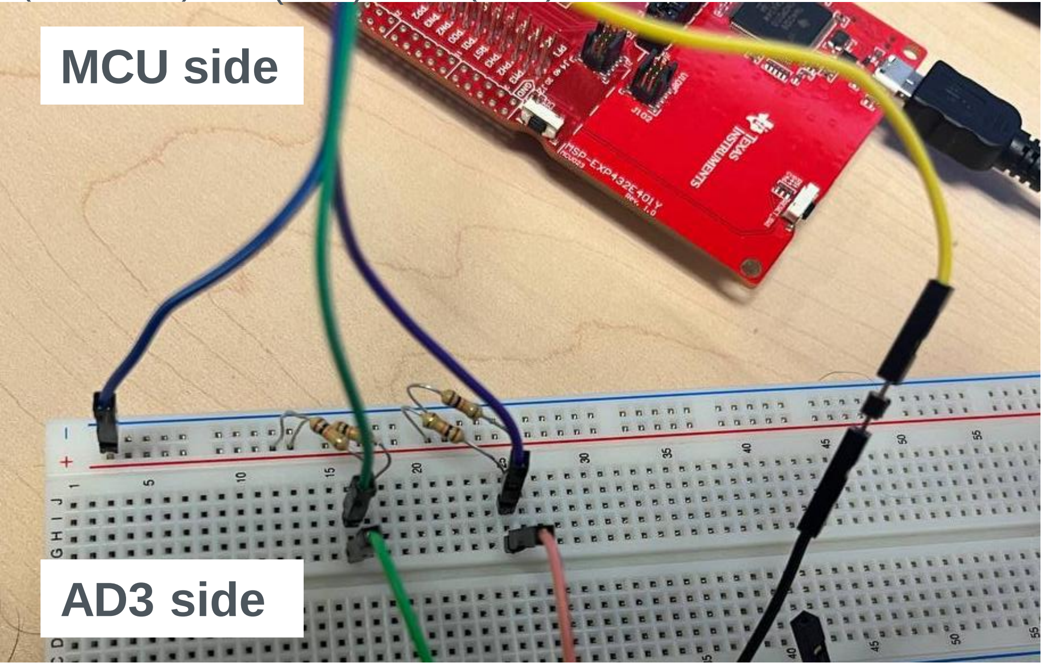
Yellow (GND)

Blue (VCC 3V3)

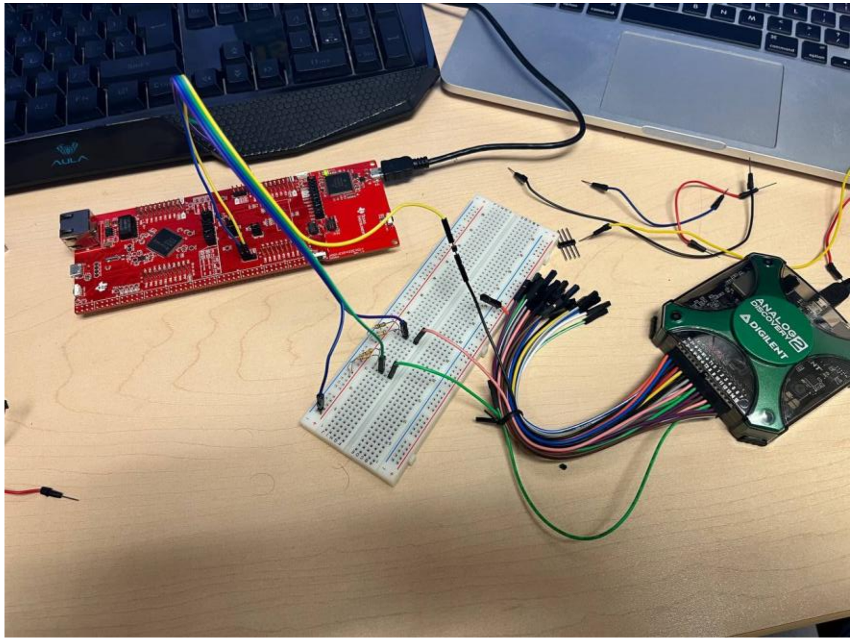


Circuit Setup

Blue (VCC 3V3) | Green (PB3) | Purple (PB2) | Yellow (GND)



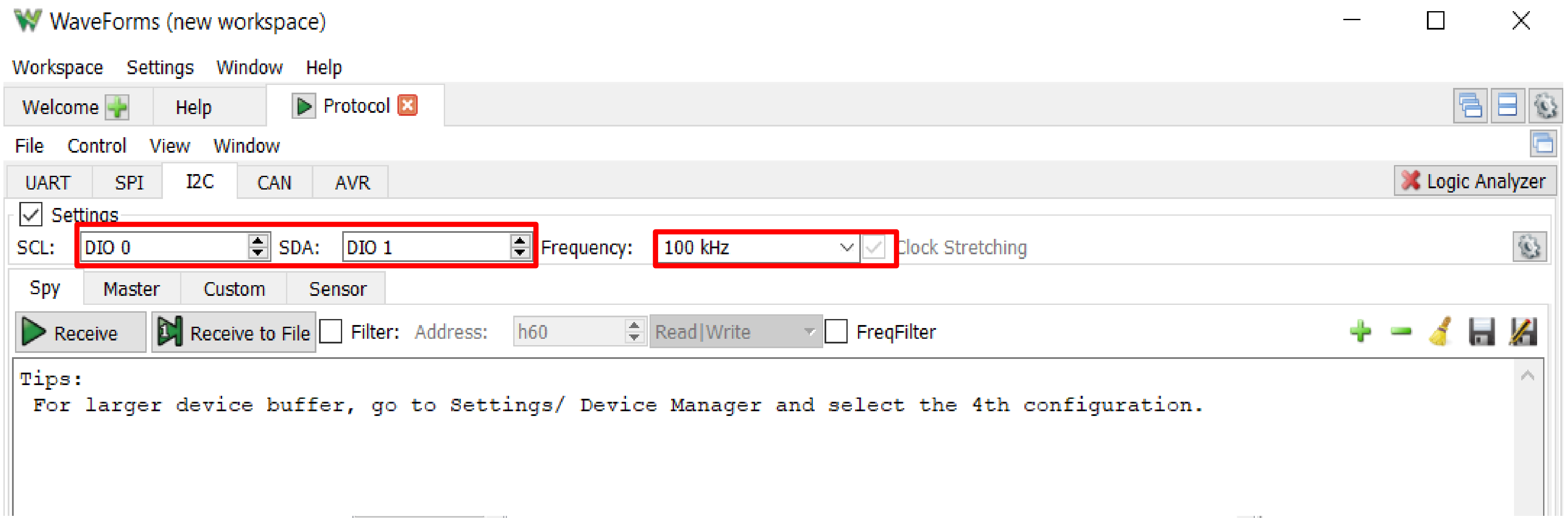
Green (DIO1) | Pink (DIO0) | Black (GND)



A view of all the wiring

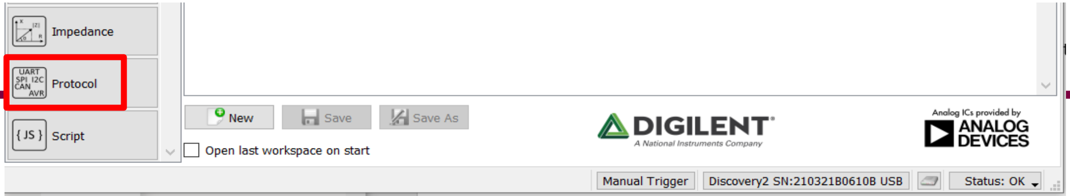
Protocol workspace (I^{2}C Packet view)

- Open **WaveForms**. From the left Panel, open **Protocol** workspace.
- Select the **I^{2}C tab**, and check the configuration to be as shown below



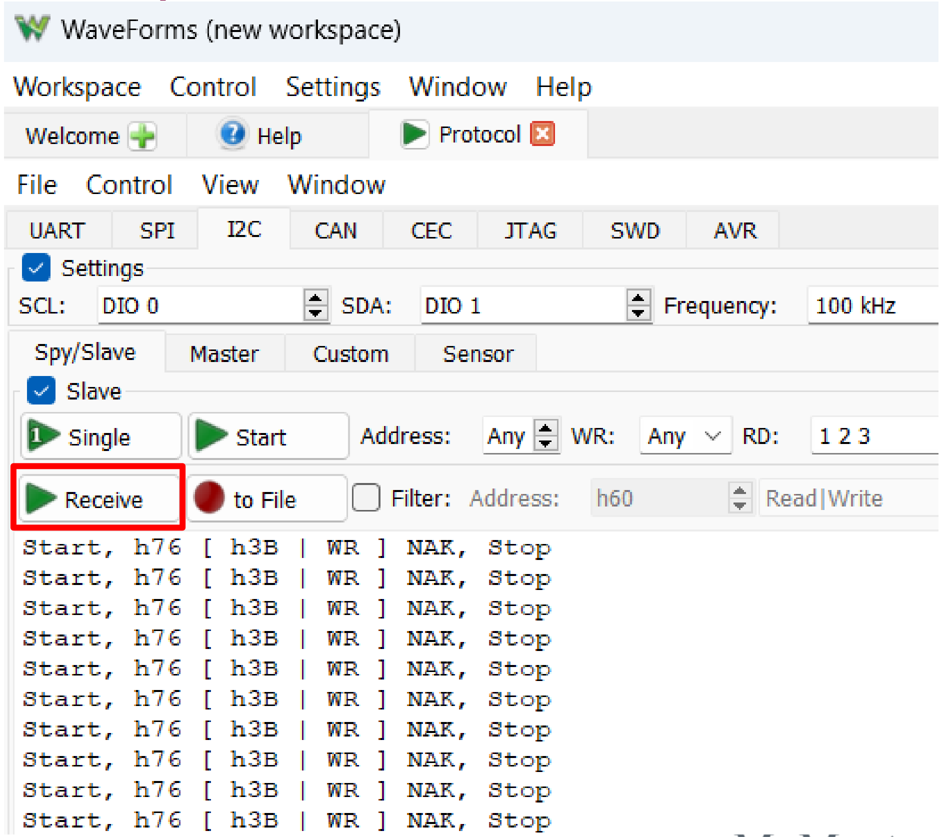
Settings
SCL: DIO 0 SDA: DIO 1 Frequency: 100 kHz Clock Stretching
Filter: Address: h60 Read|Write

```
Tips:
 For larger device buffer, go to Settings/ Device Manager and select the 4th configuration.
```


Impedance
Protocol
Script
New
Save
Save As
Open last workspace on start
DIGILENT
A National Instruments Company
Analog ICs provided by ANALOG DEVICES
Manual Trigger
Discovery2 SN:210321B0610B USB
Status: OK

Protocol workspace (I^2C packet view)

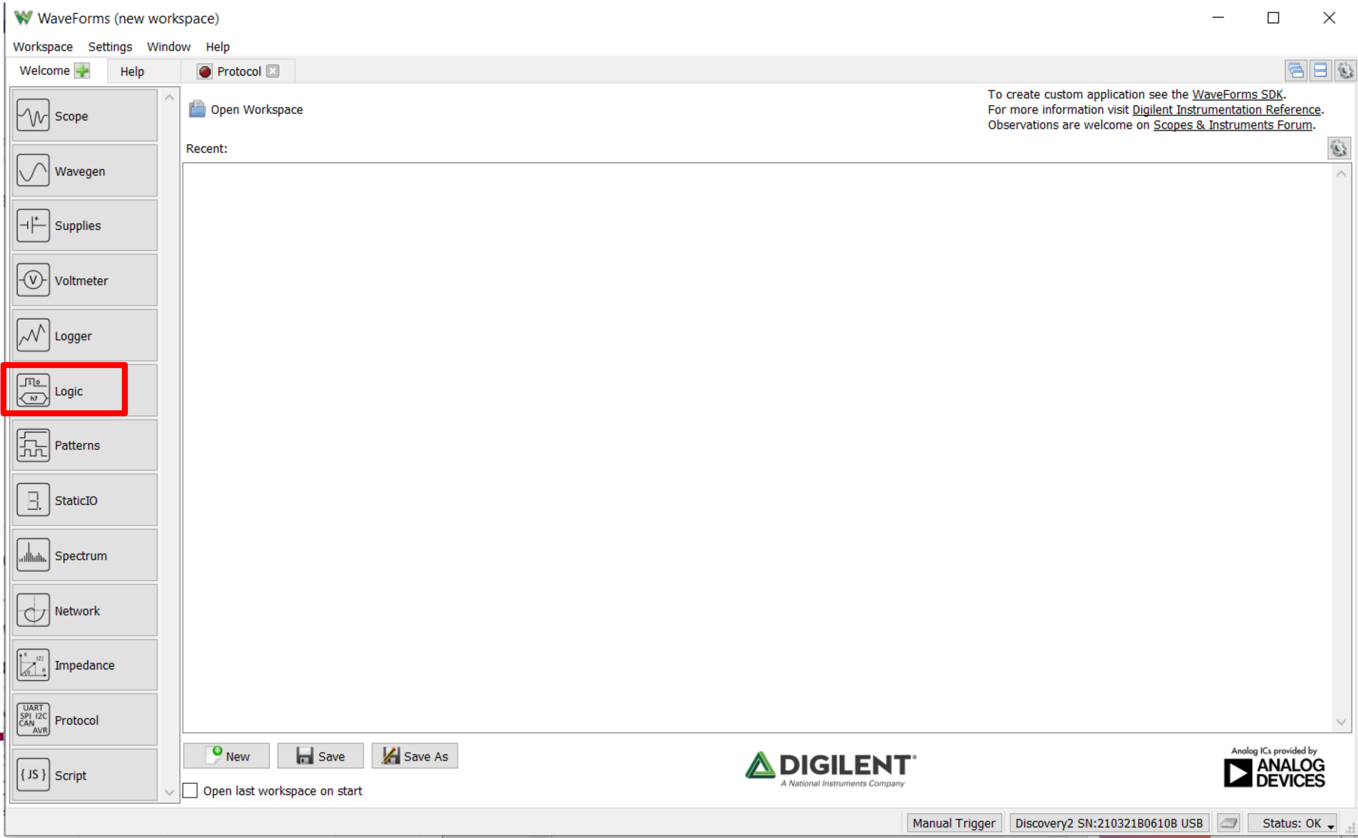
- Open the **Studio7C_I2C** project
- Translate, build and load the code on the MCU
- Run the code (press the reset button)
- On the AD3, in the I^2C **Protocol workspace**, click **Receive**



```
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
Start, h76 [ h3B | WR ] NAK, Stop
```

Logic Workspace (I^{2}C signal view)

Close the Protocol workspace and open the **Logic workspace**


WaveForms (new workspace)

Workspace Settings Window Help

Welcome | Help | Protocol

Open Workspace

To create custom application see the WaveForms SDK.
For more information visit Digilent Instrumentation Reference.
Observations are welcome on Scopes & Instruments Forum.

Recent:

New | Save | Save As

Open last workspace on start

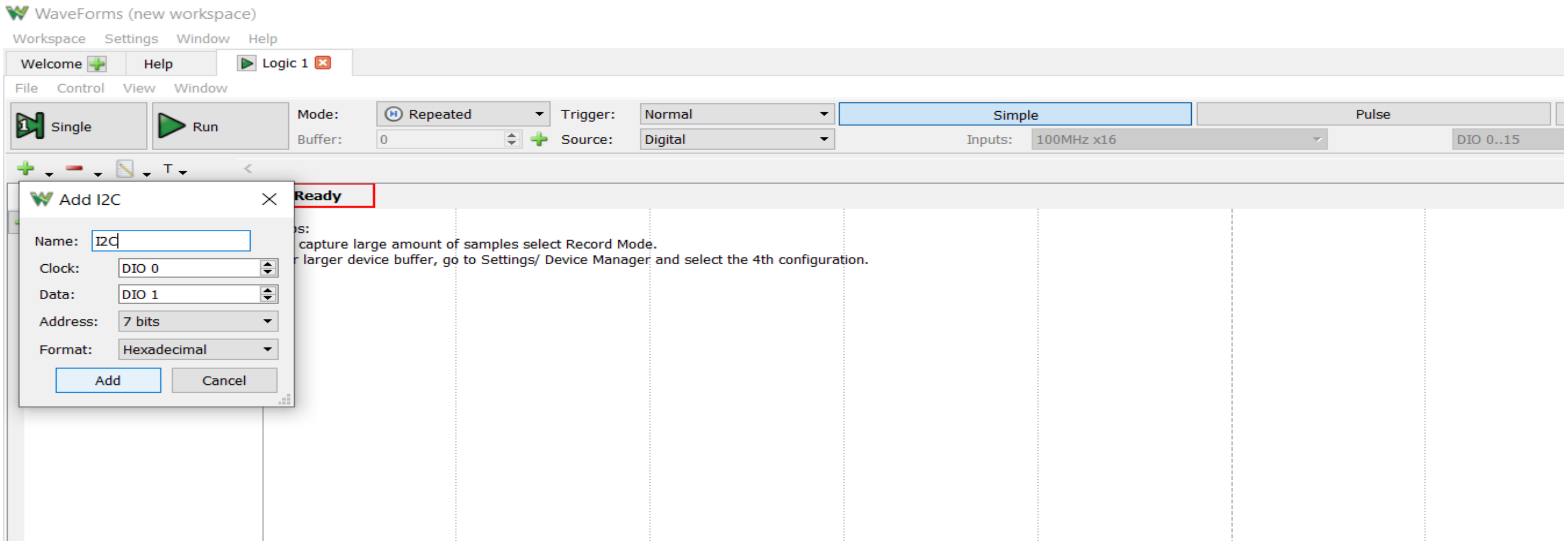
DIGILENT A National Instruments Company

Analog ICs provided by ANALOG DEVICES

Manual Trigger | Discovery2 SN:210321B0610B USB | Status: OK

Logic workspace (I^{2}C signal view)

Click on "Click to Add channels," select I^{2}C then click "Add"



Mode: Repeated
Buffer: 0
Trigger: Normal
Source: Digital
Inputs: 100MHz x16
DIO 0..15

Add I2C

Name: I2C
Clock: DIO 0
Data: DIO 1
Address: 7 bits
Format: Hexadecimal

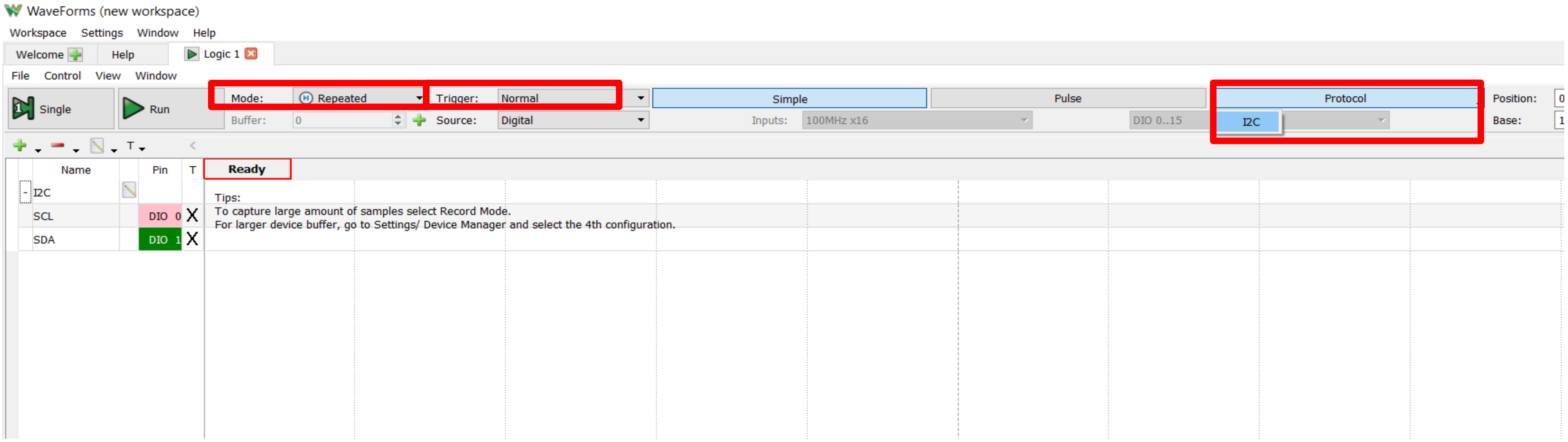
Add Cancel

s:
capture large amount of samples select Record Mode.
r larger device buffer, go to Settings/ Device Manager and select the 4th configuration.

McMaster University

Logic workspace (I^{2}C signal view)

Check the settings below; click on Protocol and select I^{2}C



WaveForms (new workspace)

Workspace Settings Window Help

Welcome Help Logic 1

File Control View Window

Single Run

Mode: Repeated Trigger: Normal

Buffer: 0 Source: Digital

Simple Pulse Protocol

Inputs: 100MHz x16 DIO 0..15 I2C

Position: 0 Base: 1

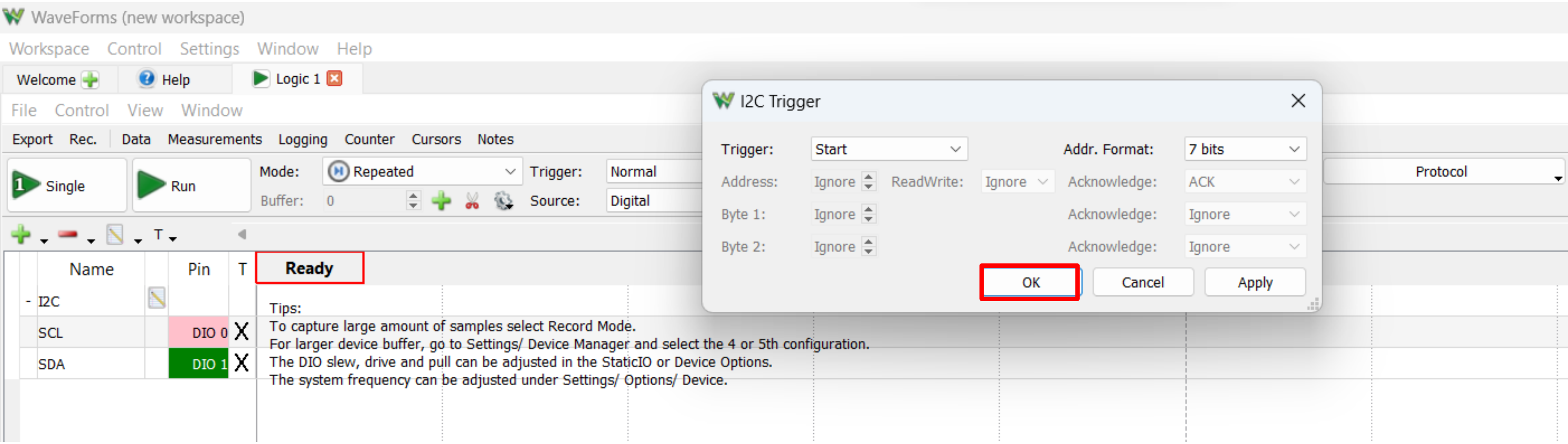
Name	Pin	T
I2C		
SCL	DIO 0	X
SDA	DIO 1	X

Ready

Tips:
To capture large amount of samples select Record Mode.
For larger device buffer, go to Settings/ Device Manager and select the 4th configuration.

Logic workspace (I^2C signal view)

Verify settings are as shown below; click OK


WaveForms (new workspace)

Workspace Control Settings Window Help

Welcome Help Logic 1

File Control View Window

Export Rec. Data Measurements Logging Counter Cursors Notes

Single Run Mode: Repeated Trigger: Normal

Buffer: 0 Source: Digital Protocol

Name	Pin	T
- I2C		
SCL	DIO 0	X
SDA	DIO 1	X

Ready

Tips:
To capture large amount of samples select Record Mode.
For larger device buffer, go to Settings/ Device Manager and select the 4 or 5th configuration.
The DIO slew, drive and pull can be adjusted in the StaticIO or Device Options.
The system frequency can be adjusted under Settings/ Options/ Device.

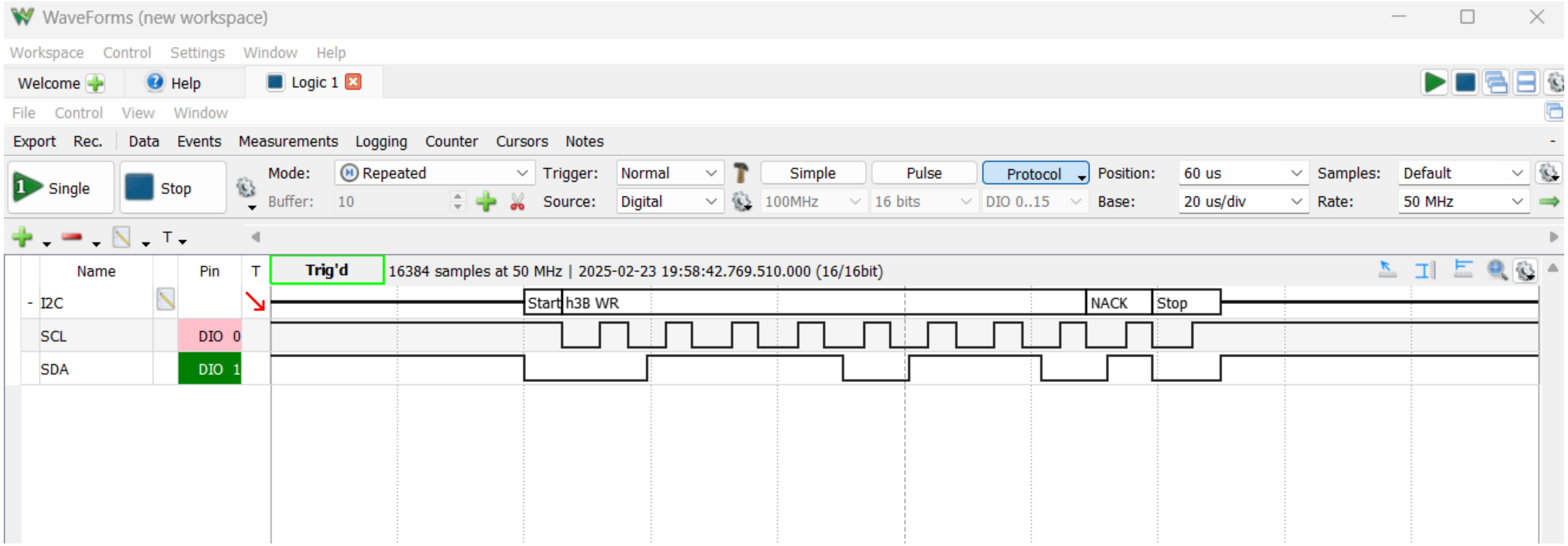
I2C Trigger

Trigger: Start | Addr. Format: 7 bits
Address: Ignore | ReadWrite: Ignore | Acknowledge: ACK
Byte 1: Ignore | Acknowledge: Ignore
Byte 2: Ignore | Acknowledge: Ignore

OK Cancel Apply

Logic workspace (I^{2}C signal view)

Click **Run** then adjust the time scale to see the signals.

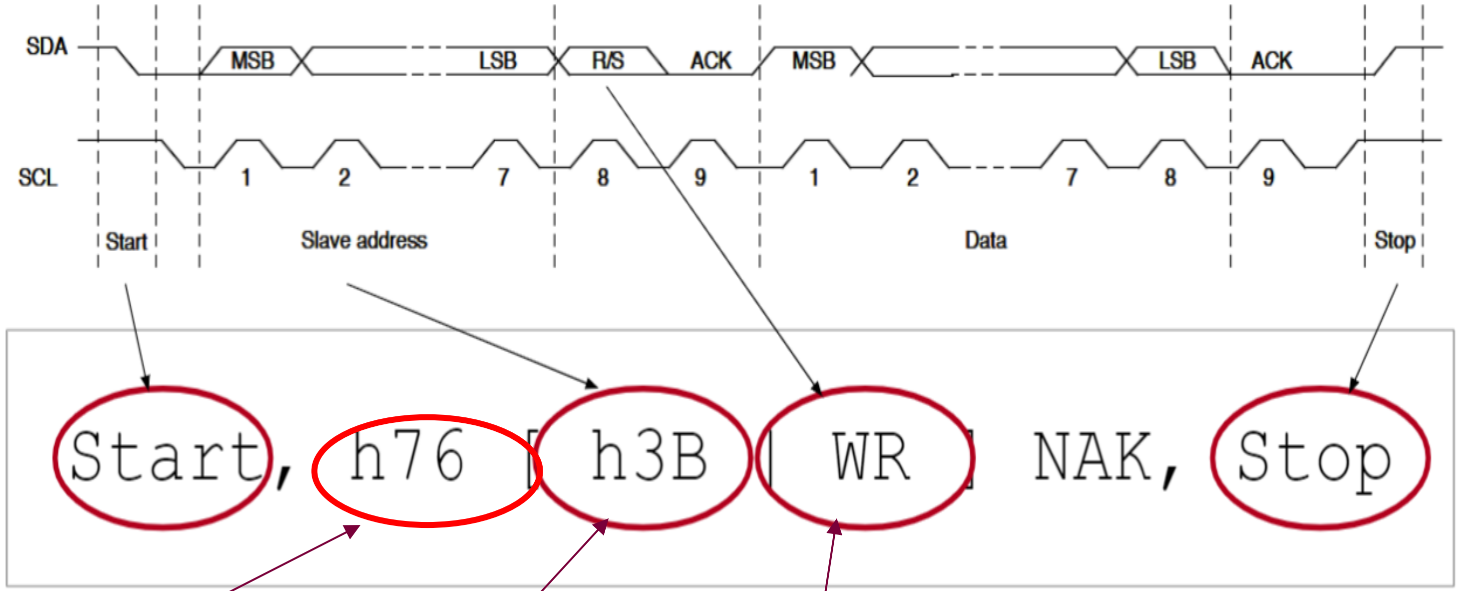

WaveForms (new workspace)
Workspace Control Settings Window Help
Welcome
Help
Logic 1
File Control View Window
Export Rec. Data Events Measurements Logging Counter Cursors Notes
Single
Stop
Simple
Pulse
Protocol
Name
Pin
T
Trig'd
I2C
SCL
SDA
Start
h3B WR
NACK
Stop

Mode: Repeated
Buffer: 10
Trigger: Normal
Source: Digital
100MHz
16 bits
DIO 0..15
Position: 60 us
Base: 20 us/div
Samples: Default
Rate: 50 MHz

16384 samples at 50 MHz | 2025-02-23 19:58:42.769.510.000 (16/16bit)

Name	Pin
I2C	
SCL	DIO 0
SDA	DIO 1

Explanation of signals (Follower not available)



The 0x76 represents:

- Follower address (7-bits) : 0x3b address of Follower peripheral
- read/write mode (1-bit). Write mode so the bit is zero in this case.

Read more about the I^{2}C protocol here
https://www.pololu.com/file/0J435/UM10204.pdf

This concludes Studio 7C

Next up:

- Studio 7D: I^2C protocol with the ToF sensor
- Project Deliverable 1 – Early Integration Demonstration (Lab 6)