Project Final Report

Wireless sensor station via radio link using auto baud rate detection

April 28, 2008
Ritu Bajpai
ritu_ece@gwu.edu

Project Abstract

This project was aimed to establish a radio link between two Z8 microcontroller boards and to transfer data between them using auto baud rate detection technique. An analog temperature sensor was connected to one of the boards and the temperature was displayed on the led array. This temperature data was to be transmitted to another Z8 board using the radio transmitter receiver pair and was to be displayed on the led array of the second Z8 board. Auto baud rate detection was to be implemented in the reception of the data. I had to decide on the auto baud rate detection algorithm to be used for this project and the suitable range for baud rate detection.

Status

The auto baud rate detection algorithm was successfully implemented within the range of permissible baud rates for the transmitter and the receiver. The transmitter receiver pair though failed to establish a radio link between the two Z8 boards as expected. The radio transmitter receiver pair worked when tested independently. The two Z8 could communicate when connected by UARTs using a wire but the radio transmitter receiver pair could not successfully establish communication between the two boards. The problem with the radio transmitter and receiver was analyzed and has been presented later in the report.

Specification

Two Z8 6401 microcontroller boards were used for this project. The analog temperature sensor was used which was interfaced through an analog to digital controller on the Z8 board. This temperature sensor is a temperature to voltage converter. The voltage output of the sensor is linearly related to the surrounding temperature.

Radiotronix transmitter receiver pairs were used in an attempt to interface the two Z8 microcontroller boards.

The code for the detection of the temperature from the sensor to the board was the same as one used in lab 3. Analog temperature sensor was used because it was easy to configure and to connect to the board. Also the pin diagram was simple as it just had three pins on the chip. The ADC was configured to convert the analog data received from the temperature sensor to digital data which could then be processed by the microcontroller to give the reading of temperature.

The code for displaying the temperature on the led arrays was also the same as one used throughout the course of this class. A library was developed to display various characters on the Z8 board led array. This was a reusable library and was appended with code for displaying new characters as per the need for each lab.

The code specially written for this project was for the transmitting and the receiving Z8 boards where the UARTs were used to transfer the data bytes and then to receive them on the other board. UART0 on Z8 board 1 was configured to talk to the UART1 of the Z8 board 2. Auto baud rate detection was implemented on the receiving side.

Equipment used:

Microchip TC1046 High voltage temperature to voltage converter
Two Zilog Z86401 Z8 Encore! Microcontroller boards
Radiotronix 433 MHz Transmitter and Receiver
RCT-433-AS Transmitter
RCR-433-RP Receiver

Implementation & Construction

Pin outs

a) Z8 1 connected to the temperature sensor and the transmitter

Analog temperature sensor
Microchip TC1046

1 VDD – Connected to pin marked VCC on the Z8 board which is 3.3 V DC supply
2 VOUT – Connected to pin marked ALG0/PB0 on the Z8 board
3 VSS – Connected to a pin marked GND on the Z8 board

Transmitter
Radiotronix
RCT-433-AS

1 ANT – Antenna, 12” long piece of wire in the kit
2 GND - Connected to a pin marked GND on the Z8 board
3 DATA – Connected to pin marked TX0 on the Z8 board
4 VCC - Connected to pin marked VCC on the Z8 board which is 3.3 V DC supply.

b) Z8 2 is connected to the receiver

Receiver RCR-433-RP

1 ANT – Antenna, 12" long piece of wire in the kit
2 GND - Connected to a pin marked GND on the Z8 board
3 GND - Connected to a pin marked GND on the Z8 board
4 VCC - Connected to pin marked VDD on the Z8 board which is a 5 V DC supply
5 VCC - Connected to pin marked VDD on the Z8 board which is a 5 V DC supply
6 Analog – Not connected
7 Data – Connected to pin marked RX1 on the Z8 board
8 GND - Connected to a pin marked GND on the Z8 board

Hardware block diagram

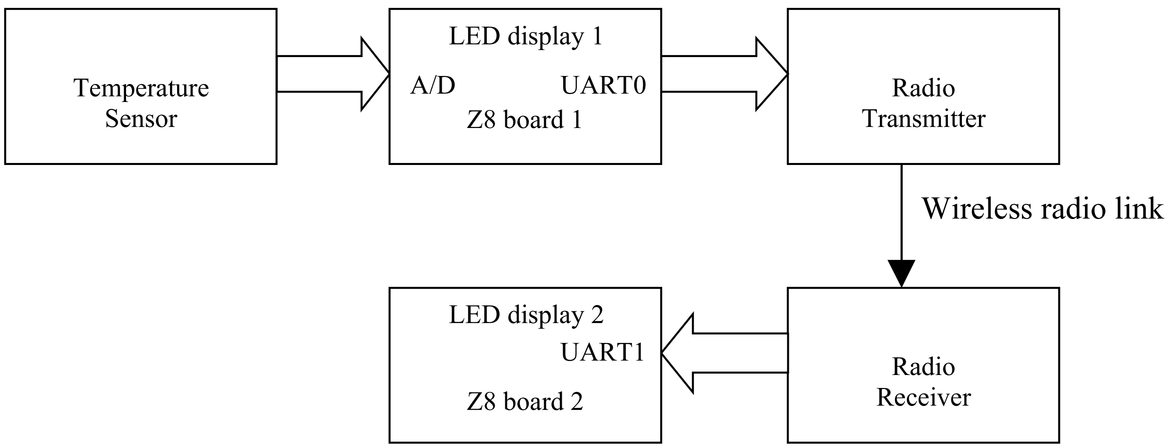

Figure 1: Hardware block diagram for the project

Flowchart of the auto baud rate detection algorithm

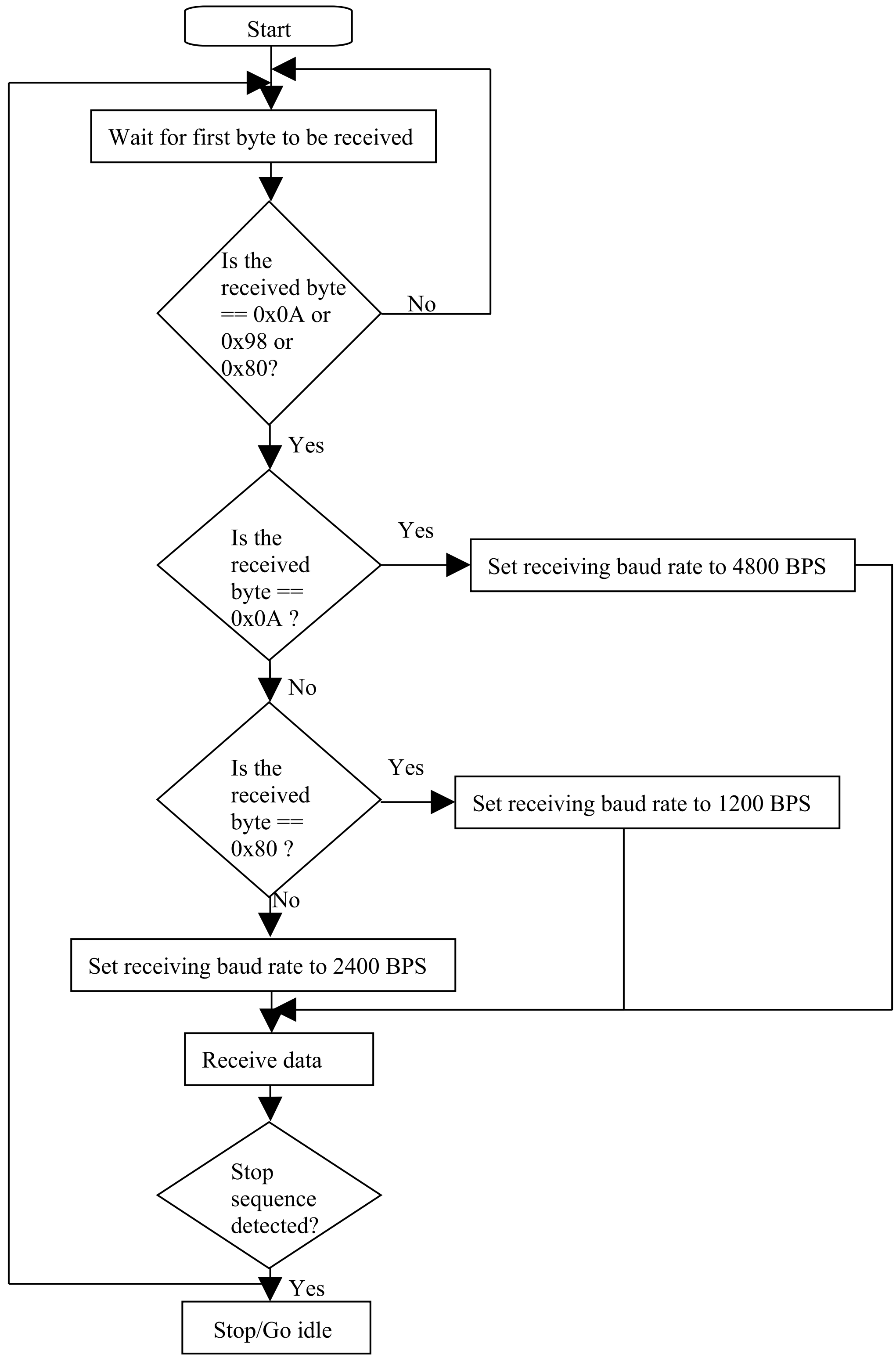

Figure 2: Flowchart for the auto baud rate detection algorithm

The flow chart for the auto baud rate detection algorithm used in this project is shown in figure 2. The byte used for the start of transmission was 0x0A which is received at the receiver starting from LSB first. Therefore 0x0A is received as follows:

Transmitter at 4800 sends 0x0A
Receiver at 4800 receives 0010100001
Receiver at 2400 receives 00001100110000000011
Receiver at 1200 receives 0000000011110000111100000000000000001111

Hence the start of transmission byte was chosen such that at all three baud rates it gave start bit as 0 and stop bit as 1 so that the UART will not detect an error in the first byte.

Also we see that for a transmitter at baud rate 4800, if start of transmission byte is 0x0A, it is received as 0x0A at the receiver when the receiver is set to baud rate 4800. On the other hand if the transmitter is set to baud rate 2400 or 1200 BPS the first byte received is 0x98 and 0x80 respectively at the receiver working at baud rate 4800 BPS.

Therefore the approach adopted in this project is to set the receiver at 4800 baud rate and to wait for the first byte. Depending on the first byte received, the baud rate of the receiver is then set to 4800, 2400 or 1200 BPS.

To determine the baud rates with which I can work I had to look into the datasheets of the transmitter and receiver pair that I was going to use.

- Part used
  RCT-433-AS Radio transmitter
  Baud Rate – NRZ
  Min DC
  Max 4800 BPS

- Part used
  RCR 433 RP Radio receiver
  Baud Rate – NRZ
  Min 1200 BPS
  Max 4800 BPS
  Baud Rate – PWM
  Min 120

  Max 2400 BPS

Therefore the compatible range for working can be NRZ 1200-4800 BPS. I wanted to fix the receiver baud rate to 4800 and change the transmitter baud rate to work at 1200, 1800, 2400 or 4800 baud rate. Later I realized that 1800 baud rate is not permissible for the UART on Z8 board. Therefore I worked with baud rates 4800, 2400 and 1200 BPS for the transmission.

Software block diagram

Transmitter

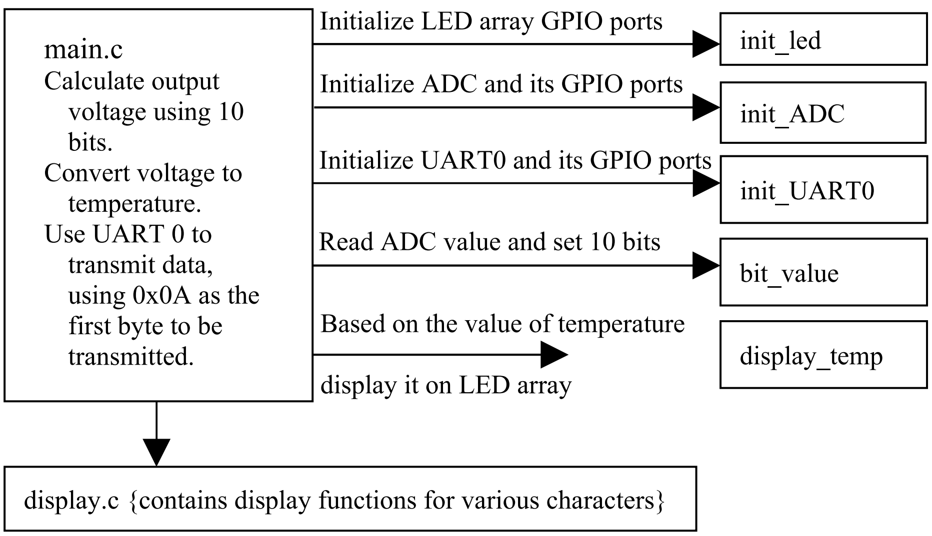

Figure 3: Software block diagram for the Z8 configured as the transmitting side.

Receiver

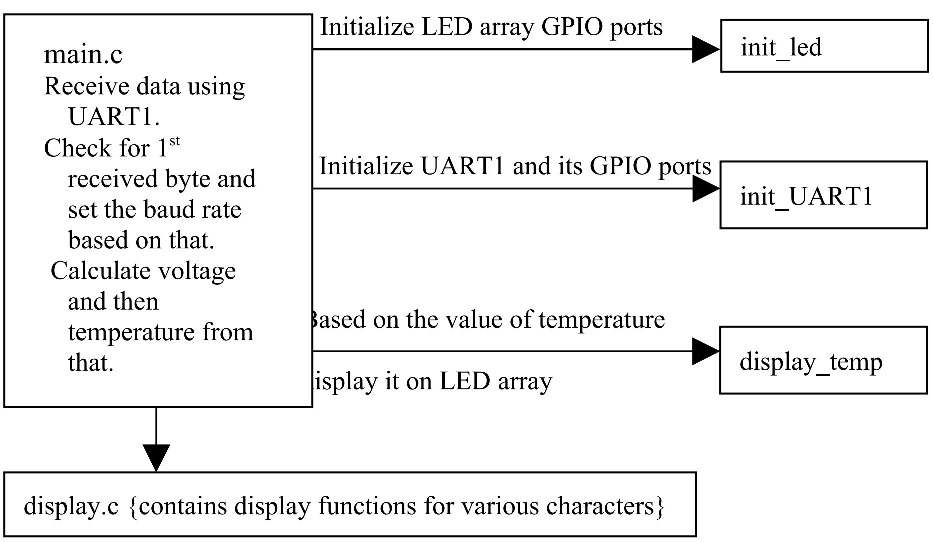

Figure 4: Software block diagram for the Z8 configured as the receiving side.

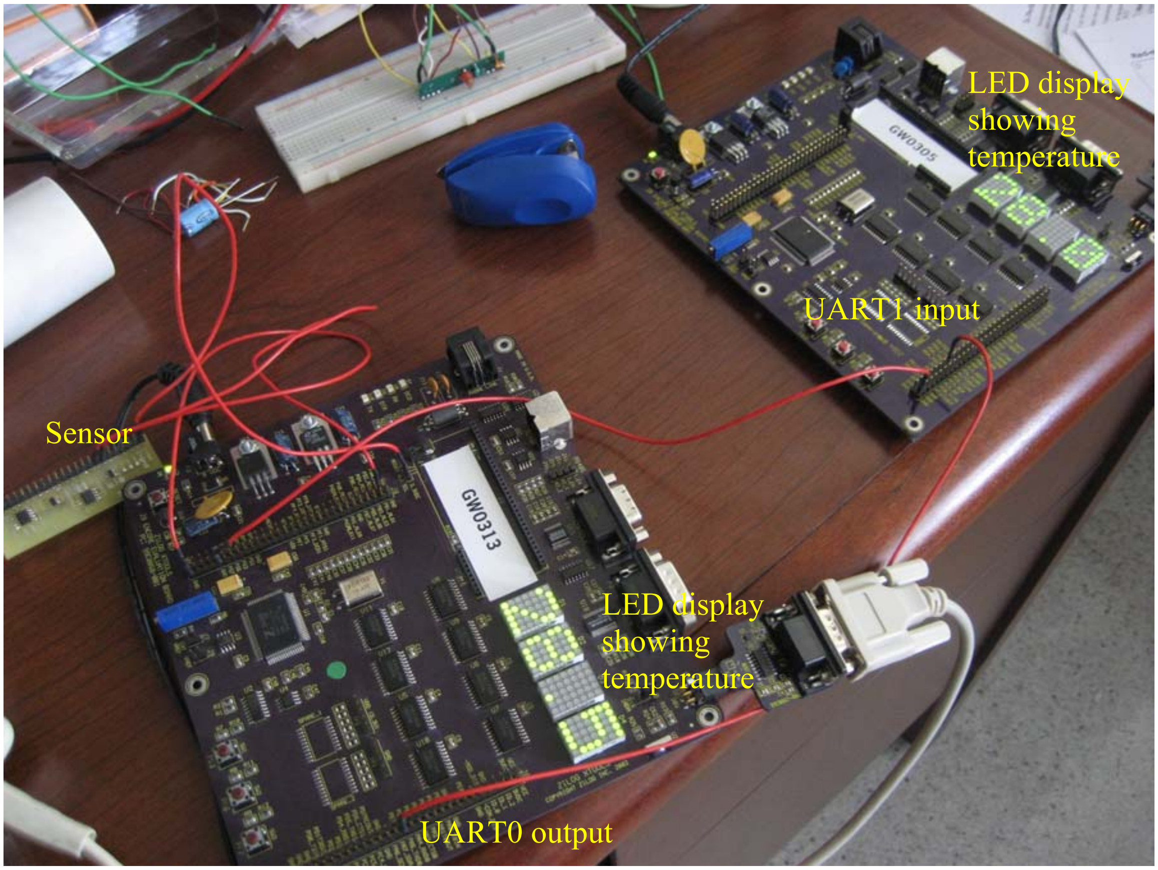

Figure 5: Picture of the project showing the two boards connected to each other using UARTs. The two UARTs are connected by a wire. Sensor is connected to the Z8 on the left.

Figure 5 shows the picture of the project with the sensor connected to the Z8 board 1 (on the left). The temperature is displayed on the LED array. The temperature sensor gives the analog voltage value corresponding to the temperature. This value is sent to the ADC which converts it to digital data. This digital data is directly sent to the UART0 and is taken as an output at pin TX0 and is fed by a wire to the UART1 at pin RX1 on the Z8 board 2 (on the right). The received data is then processed by the microcontroller to give the value of the temperature. The temperature reading is displayed on the LED array similar to Z8 1.

The system was then tested to show that when the surrounding temperature in changed the temperature reading on both the displays should change simultaneously. This was successfully demonstrated and hence it was sure that the two UARTs were properly configures and the two Z8's were communicating properly.

Once this was done the wired connection was removed and the radio transmitter was connected to pin TX0 on the Z8 1 board and the radio receiver was connected to the pin RX1 on the Z8 2 board. The transmitter and the receiver failed to communicate and therefore I decided to separately test the transmitter and receiver pair using the oscilloscope.

Debugging the radio transmitter and the receiver problem

Step 1: Checking if the TX0 pin on Z8 board one was giving the right data and the RX1 pin on the Z8 board 2 was able to read it properly and pass it to UART.

Both pins were connected by wire and data 0x0D was transmitted.

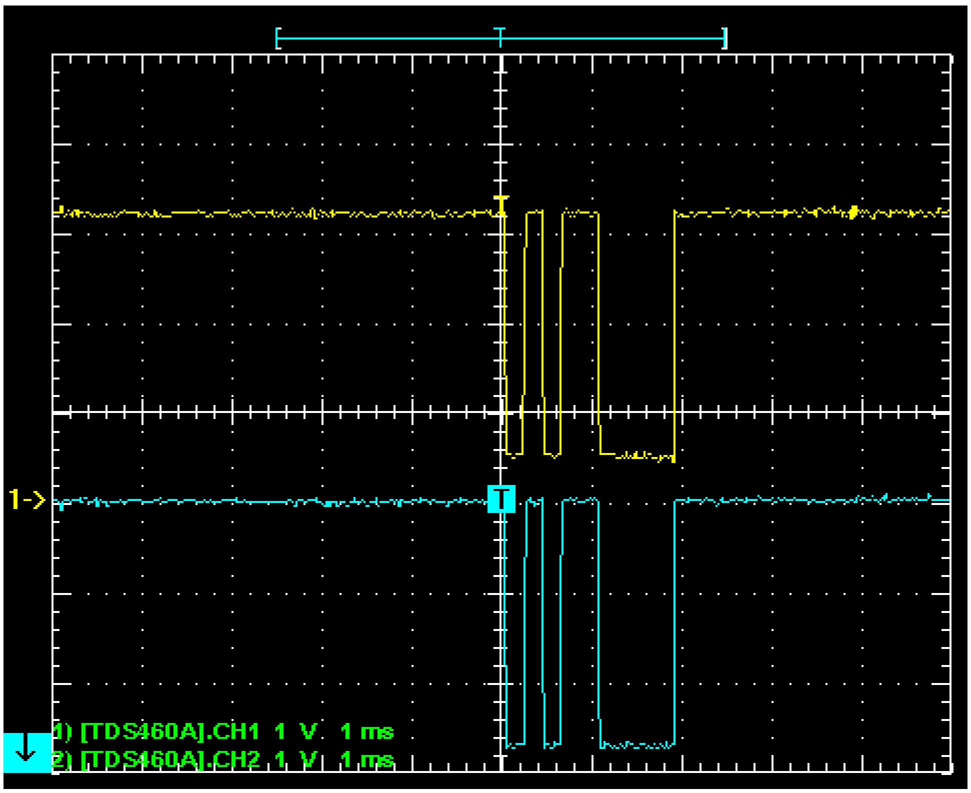

Figure 6: Output waveform when both boards were connected by wire. Yellow TX0 on board 1 and blue RX1 on board 2.

The waveform showed that the Z8's were configured correctly and were able to communicate via UARTs.

Step 2: Checking if the radio transmitter and receiver pair were working correctly. To test this, an input from a function generator was applied to the transmitter and the output waveform at the receiver was recorded on the oscilloscope.

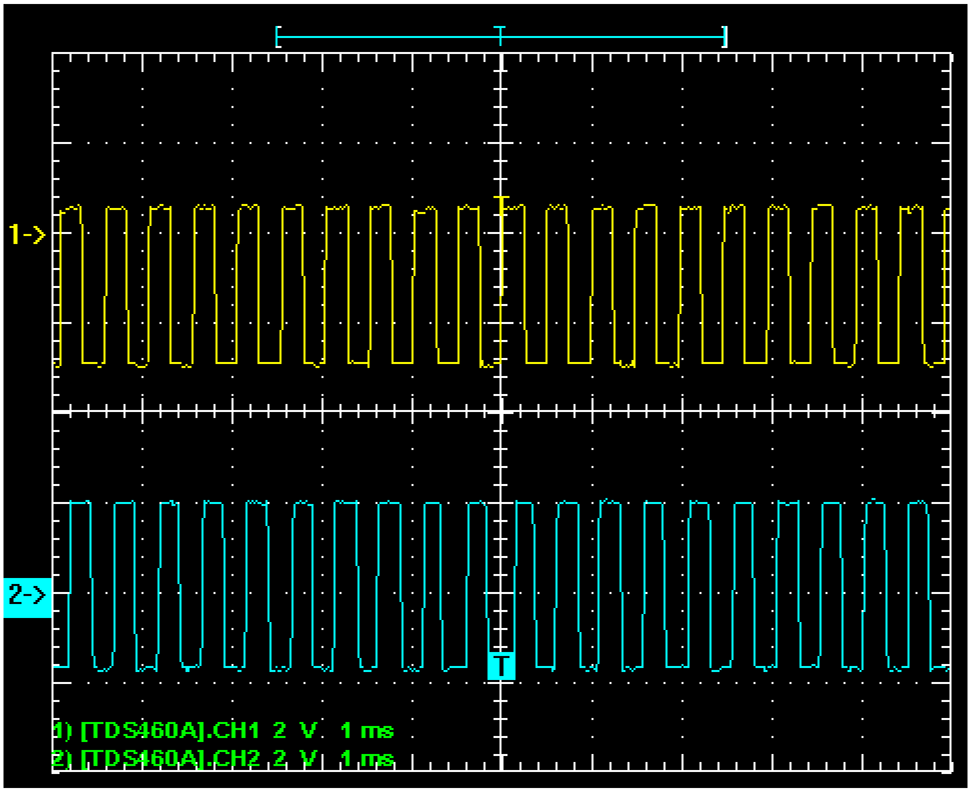

Figure 7: Testing the transmitter receiver pair with the help of a function generator. Yellow: input from the function generator, blue: output at the receiver. Signal frequency is about 3kHz.

The above test confirmed that the radio transmitter and receiver pair was working properly on its own.

Step 3: Still keeping the input to the transmitter as coming from the function generator the output of the receiver was connected to RX1 on Z8 board 2.

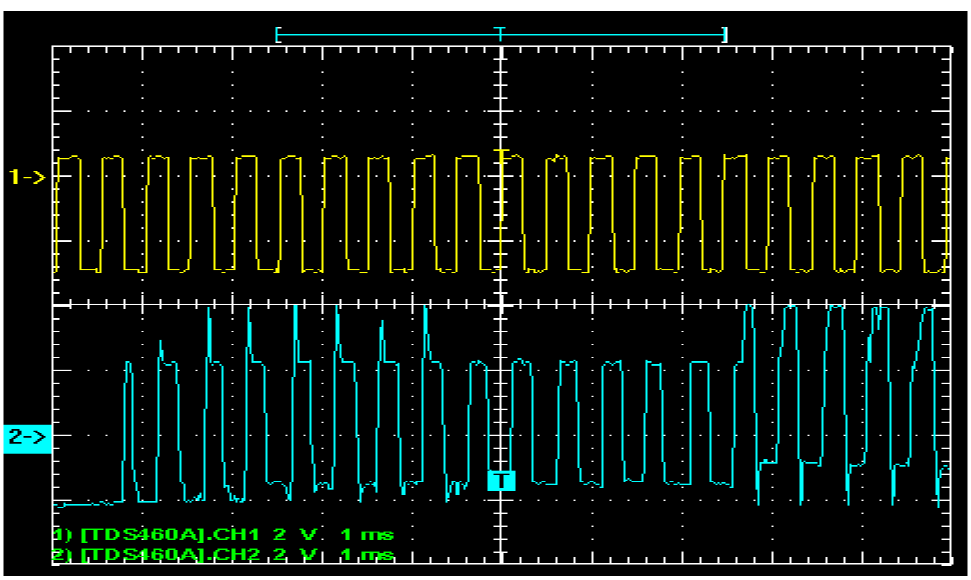

Figure 8: Input from the function generator, transmitted to receiver and connected to Z8 pin RX1.

Connecting Z8 pin RX1 to the receiver introduced noise as seen in the waveform. This is after the output was improved by increasing the length of the antenna. Still at least the bit pattern in maintained.

Step 4: Function generator input was disconnected and the input was taken from the TX0 pin from Z8 board.

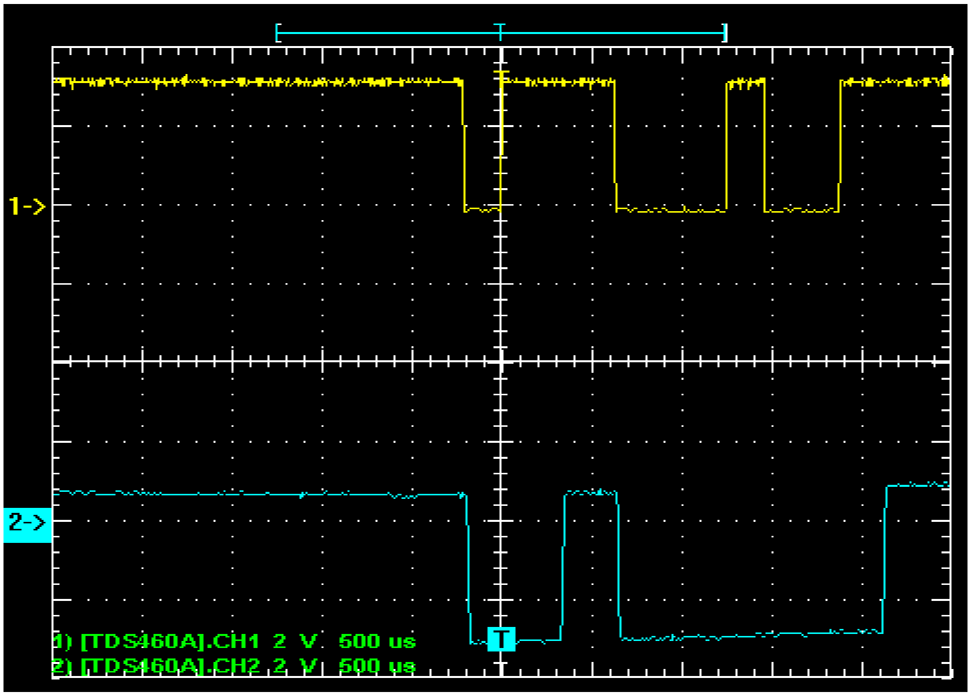

Figure 9: Input from the TX0 pin on Z8 board 1 and output at the receiver. Yellow: transmitter, blue receiver.

The bit pattern was not preserved. More than one bit pattern transmission was observed

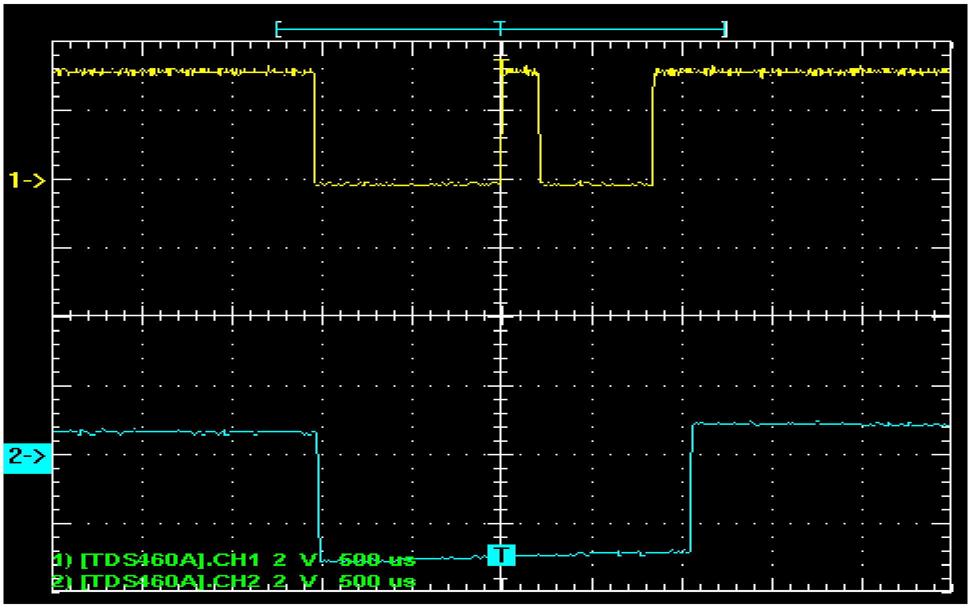

Figure 10: Another bit pattern transmission from transmitter to receiver.

Seems like 2 one bits are treated as one and single 1 bit is treated as zero. But no!

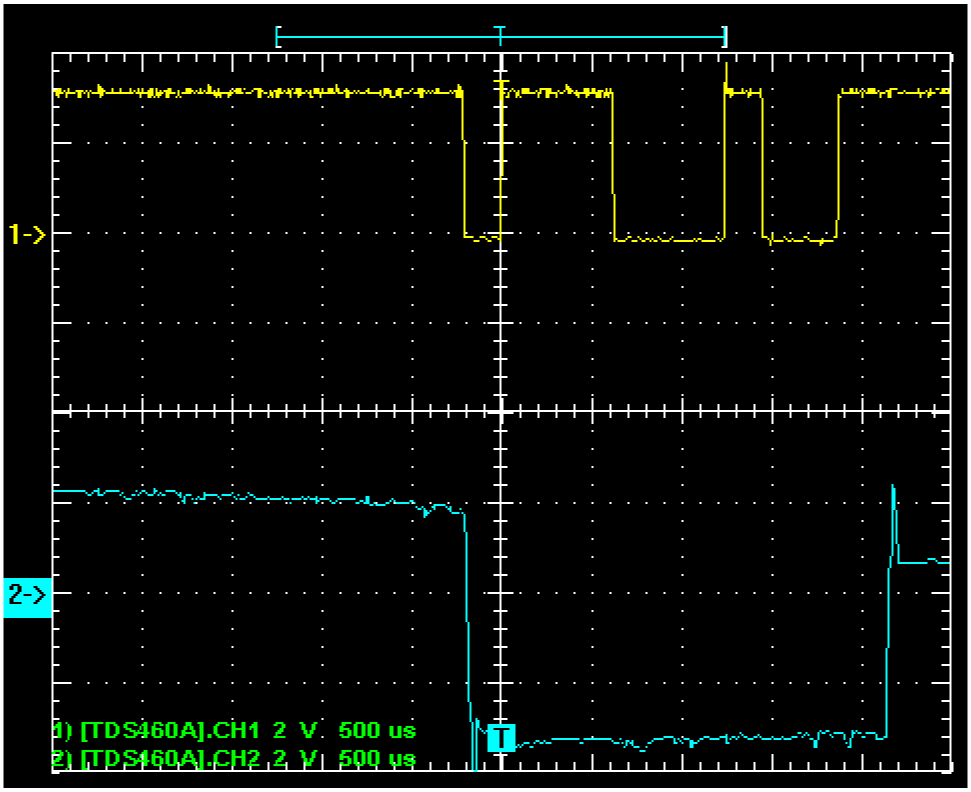

Figure 11: Another bit pattern transmission from transmitter to receiver.

Retrospective

The important part to begin the project was to decide how I am going to configure two boards to communicate with each other. I decided to use the UARTs so that they could take care of the way the data was to be transmitted in packets and I did not have to worry about making my own transmission protocols. This proved to be very beneficial as I could save a lot of time for doing the other parts of my project.

I think it was also a good idea to set up both the boards and to test that they could communicate with each other and then put the radio transmitter and receiver to send and receive the data. Since my radio transmitter and receiver pair were not successful in establishing the radio link between the boards at least I was sure at this stage that it was not my code or the microcontroller which was giving me problems but the external hardware that I was attaching with it.

Modular testing proved to be very helpful. After testing my boards and successfully having them to communicate, I tested the transmitter receiver pair independently, which also proved to work fine. The problem was occurring when I was trying to use the transmitter receiver pair with my Z8 boards. Hence modular testing helped me to pin point the problem.

I think my project was overall successful as I could make the Z8 board work the way I wanted and could also implement auto baud rate detection. So even if I was to start all over again, I think I would approach the project the same way other than the fact that I would keep at least 1 week to try and figure out what was going wrong when I was putting the transmitter receiver pair and my Z8 boards together. I think I could have come up with some hardware circuits, which could have helped to establish a radio link or to find a better pair of commercially available transmitter receiver pair. Although I do not consider this to be a failure of this project because for this class this was not the objective of the project. The transmitter receiver pair should just have worked when they were connected properly to the pins of the Z8 boards.

Attachments

- Radio transmitter and receiver datasheets.
- The analog temperature sensor data sheet.
- The source code for the transmitter and receiver.
- Brochure