

Wireless Protocol Design Documentation

**Andrew Burian
Chris Holisky
Shane Spoor
Ashley Tham**

1/12/2013

Table of Contents:

Task Breakdown	2
State Chart Diagram	3
Gantt Chart	4
Pseudo Code.....	5
Tests.....	9

Wireless Protocol Project Task Breakdown

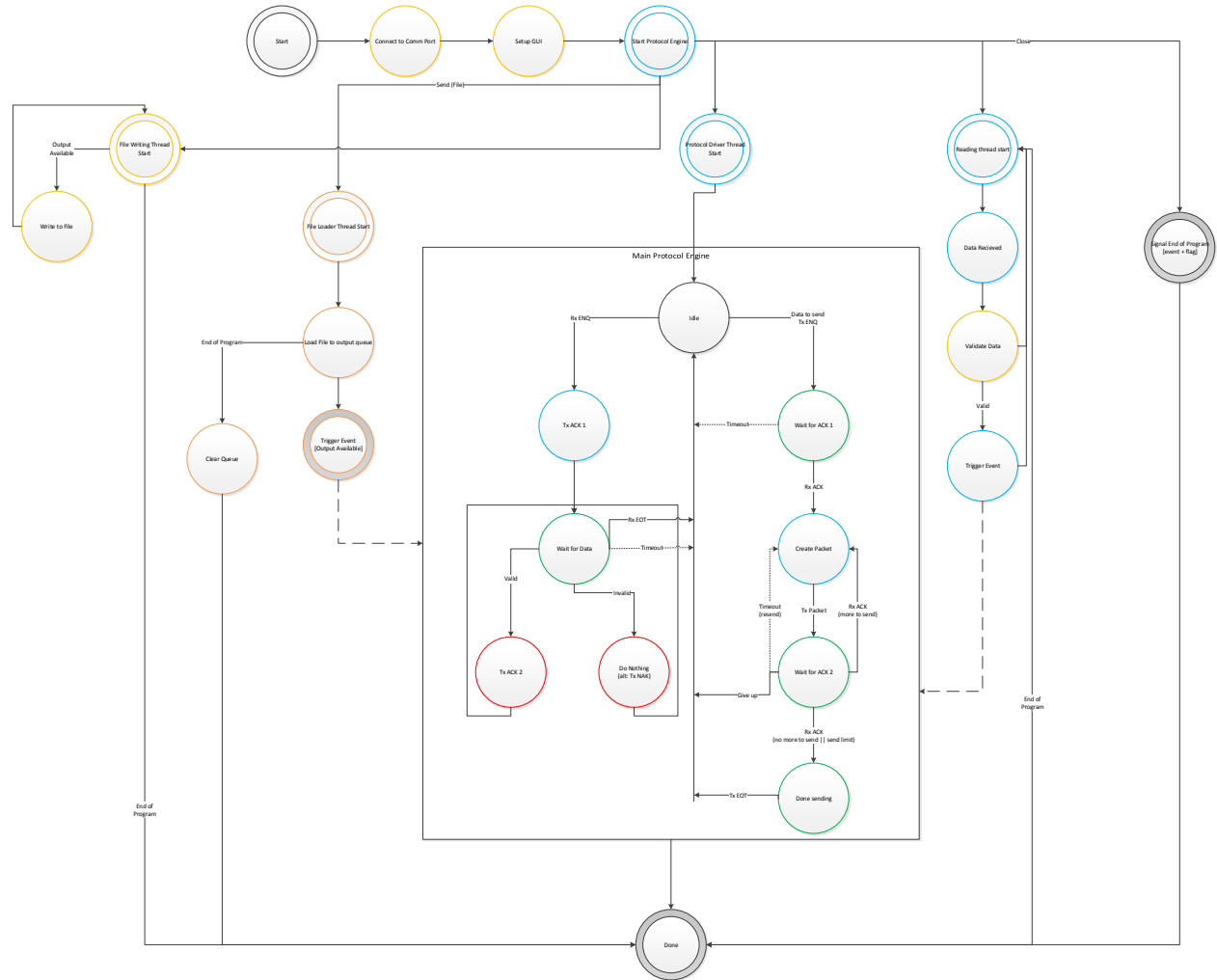
Documentation

Gantt chart	Ashley Tham
State chart diagram	Andrew Burian
Pseudo code	All Team Members
Tasks document	Ashley Tham
Compile documentation	Ashley Tham

Programming

Connect to Comm Port	Chris Holisky
Start Protocol Engine	Andrew Burian
File Loader Thread Start	Chris Holisky
Load file to output queue	Chris Holisky
Clear queue	Chris Holisky
Reading Thread Start	Andrew Burian
Data Received	Andrew Burian
Validate Data	Chris Holisky
Trigger event	Andrew Burian
Idle	Shane Spoor
TxAck(1)	Andrew Burian
Wait for Ack(1)	Shane Spoor
Create packet	Andrew Burian
Wait for Ack(2)	Shane Spoor
Done Sending	Shane Spoor
Wait for Data	Shane Spoor
TxAck(2)	Ashley Tham
Do Nothing	Ashley Tham
File Write Start	Chris Holisky
Write to File	Chris Holisky

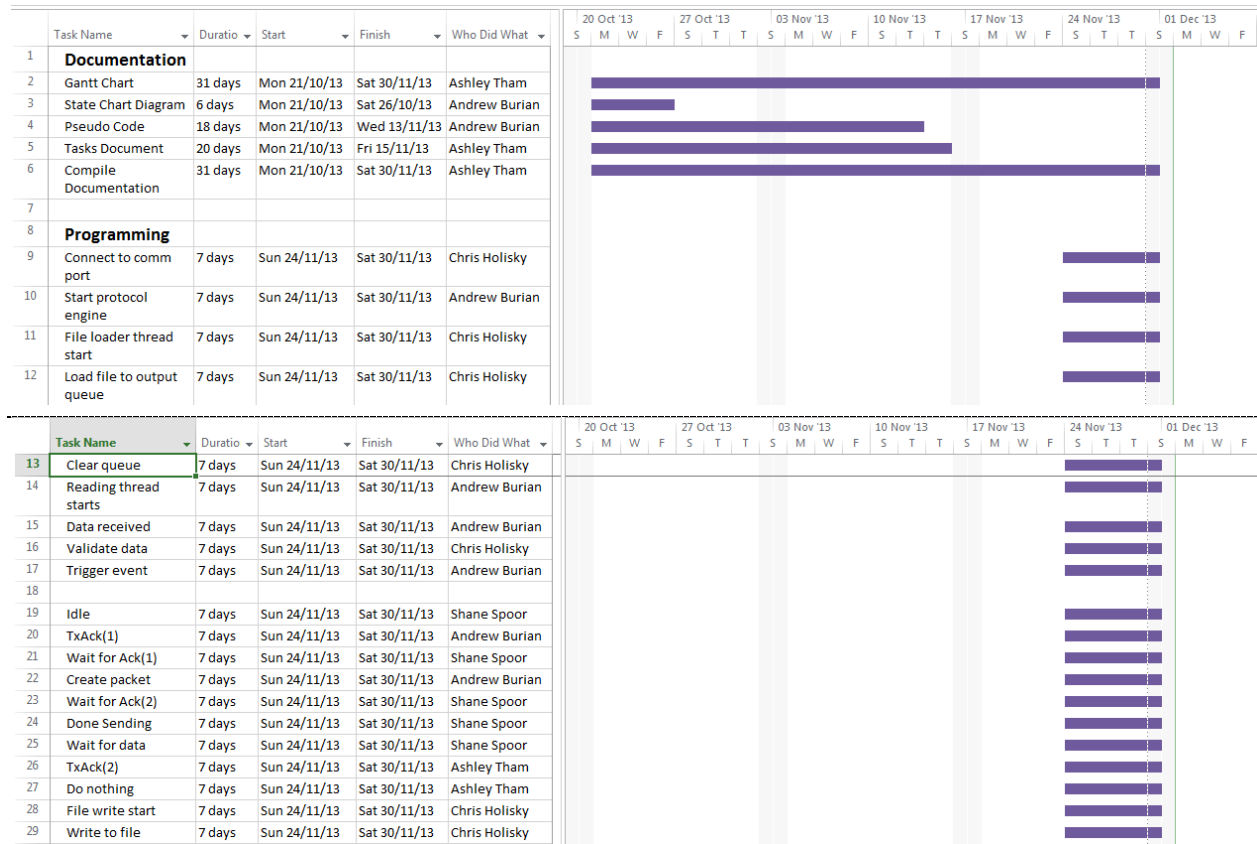
Wireless Protocol Project State Diagram



Legend:



Wireless Protocol Project Gantt Chart



Wireless Protocol Project Pseudo Code

ProtocolDriver

While the Program isn't done

 Check if there's output from incomplete previous transfer

 If there is, set Output Available event to signaled state

 Wait for either ENQ Rx'd or Output Available event

 If ENQ Rx'd

 Call RxProc

 Else if Output Available

 Call TxProc

RxProc

Tx ACK1

While True

 Wait for Data, Bad Data or EOT to be Rx'd or the program to end

 If Rx EOT or connection timed out

 Return to Idle

 Else If Rx Data

 Tx ACK2

 Else If Rx Bad Data

 Tx NAK

TxProc

Tx ENQ

 While there's data to send, the sending limit isn't reached, and retransmit hasn't failed

 Create Packet

 Tx Packet

 If response times out

 While Retransmission Attempts <= 5

 Attempt to Retransmit Packet

 If more than 5 Retransmission Attempts

 Retransmit failed; exit loop

 If Retransmit Failed

 Return to Idle

 Else

 Tx EOT

FileWriterThread

While the program isn't done

 Wait for either input to be available or the end of program event

 If the end of program event is received, exit

 Else If Input is Available

 While the input queue isn't empty and we've processed <= 1022 chars

 Copy the current character to the buffer

 Pop the current character from the input queue

 Display the contents of the buffer

FileBufferThread

Open the file

If file creation fails, then exit the thread

Read the entire file into a buffer

Copy the file contents into the output queue and signal that output is available

Free the temporary buffer

VOID ClearOutputQueue()

```
{
    WaitForSingleObject(hOutputLock, INFINITE);
    while(!(outQueue->empty()))
    {
        outQueue->pop();
    }
    ReleaseMutex(hOutputLock);
}
```

ClearOutputQueue

While the output queue isn't empty

 Remove the current char from the queue

FillDataFrame

```
{
    if SYN
        get next byte
        if Ctrl
            go to trigger event (ctrl)
        if SOT
            get next 1020 bytes
            if not duplicate
                go to validate data
            go to trigger event (dup)
```

```
}
```

```
SendNext
```

```
-----
```

```
{  
    if new  
        Add SYN  
        Add correct SOT char  
        Add data from queue  
            (pad if necessary)  
        Add CRC  
        Send packet  
        Go to Wait for Ack(2)  
    if resend  
        Send last packet  
}
```

```
CheckCRC
```

```
-----
```

```
{  
    Check crc section of data against the data packet  
    return true or false for valid data  
}
```

```
ReadCtrl
```

```
-----
```

```
{  
    If ctrl  
        trigger corresponding ctrl event  
    If dup  
        trigger data received event  
    if data-invalid  
        trigger baddata event  
    if data-valid  
        trigger data event  
        read data to input queue  
  
    return to Thread Start  
}
```

```
ProtocolDriver
```

```
-----
```

```
While the Program isn't done
```

```
    Check if there's output from incomplete previous transfer  
        If there is, set Output Available event to signaled state  
    Wait for either ENQ Rx'd or Output Available event  
        If ENQ Rx'd  
            Call RxProc  
        Else if Output Available  
            Call TxProc
```


RxProc

Tx ACK1

While True

 Wait for Data, Bad Data or EOT to be Rx'd or the program to end

 If Rx EOT or connection timed out

 Return to Idle

 Else If Rx Data

 Tx ACK2

 Else If Rx Bad Data

 Tx NAK

TxProc

Tx ENQ

 While there's data to send, the sending limit isn't reached, and retransmit hasn't failed

 Wait for End of Program, an ACK to be received, a NAK to be received, or an ENQ to be received

 If an ACK was received

 Send next packet

 Else If Response times out or NAK was received

 If the last packet sent was the ENQ

 Return to ProtocolControlThread

 Else

 Attempt to retransmit 5 times

 If more than 5 Retransmission Attempts

 Retransmit failed; exit loop

 Else If an ENQ was received

 Sleep for a random period

 If Retransmit Failed

 Return to Idle

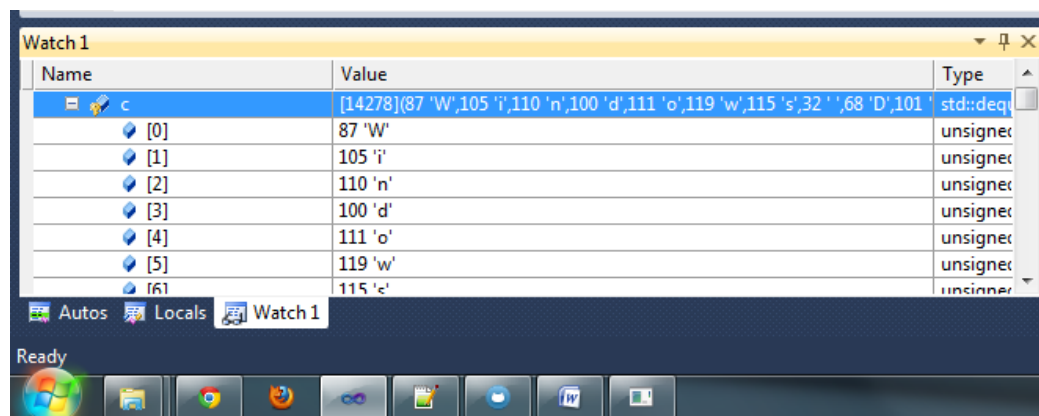
 Else

 Tx EOT

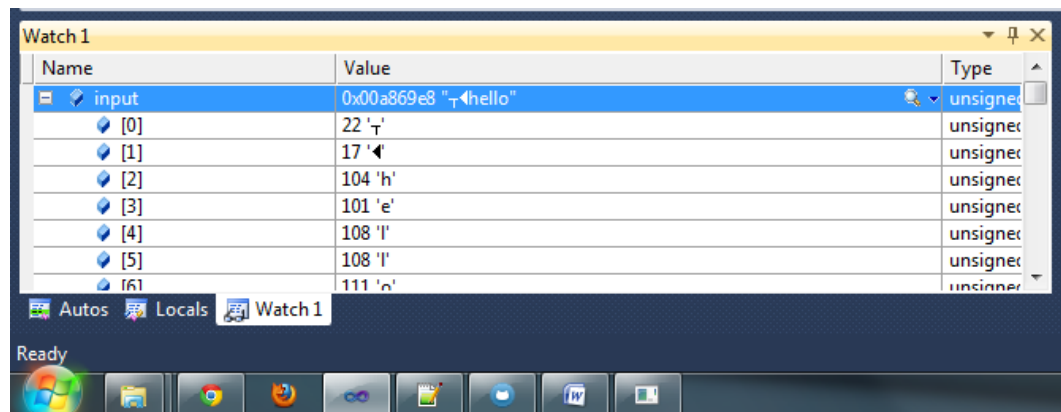
[illegible]

Number	Test	Tools Used	Expected	Pass Fail	Ref
1	File Loading to Buffer	Visual Studio Debugger	Complete File In Queue	PASS	1
2	Input Read From Comm	Visual Studio Debugger	SYN-SOT-data in input buff	PASS	2
3	Received Data Ack'd and displayed	BCP Program	Data Fully Displayed	PASS	3
4	Sent Data Ack'd and Received	BCP Program	Ack received after data sent	PASS	4
5	Wireless Retransmissions	BCP Program	Failed data resent to limit	Buggy	5

Ref 1



Ref 2



Ref 3

hello

Connect

Send File

Packet stats

Sent:0

Lost:0

Received:1

Received Corrupt:0

Idle

Ref 4

hello

Connect

Send File

Packet stats

Sent:14

Lost:0

Received:1

Received Corrupt:0

TX - pkt

Ref 5

hellohello

Connect

Send File

Packet stats

Sent:	16
Lost:	2
Received:	2
Received Corrupt:	0

TX - pkt