WX_SPIN_Text_Page_To_Micro Iss1

The objective of this project is to assist SPIN coders to access the WX ESP8266 module. Please refer to Parallax Learn tutorials for comprehensive resources and tutorials.

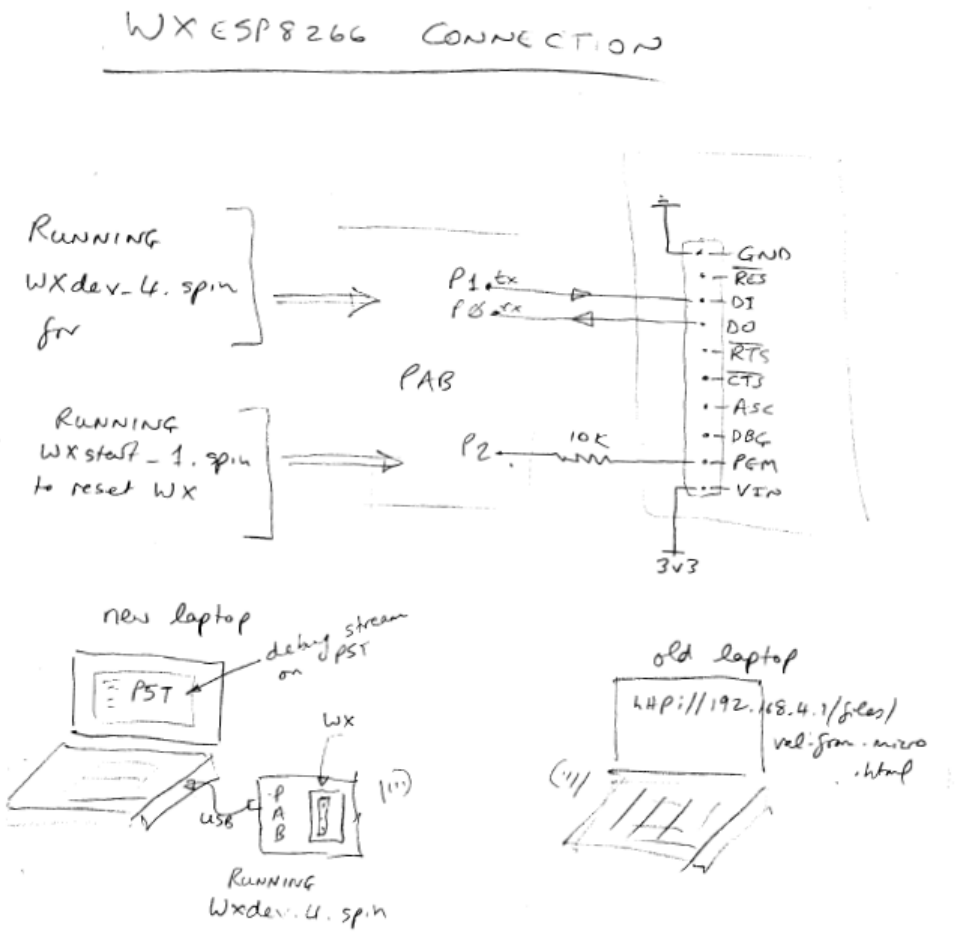
1. Schematic

Prop-connected Laptop “new laptop” run Text_Page_To_Micro_1.SPIN

Browser Laptop “old laptop” connect to WX wi-fi and load webpage

http://192.168.4.1/files/text-page-to-micro.html

as below



2. Prop-connected Laptop run Text_Page_to_Micro_1.SPIN

- Connected to PAB with WX as per schematic above
- First of all reset PAB on/off. Especially if previously running (otherwise it will open listener/handle 2)
- Run Text_Page_to_Micro_1.SPIN -> F10/F12 -> to load PST, then “m” or any key to kick off the listen command

- First of all, the LISTEN command with some API snippets;

```
m
A2_Listen command echo follows; þLISTEN:HTTP,/tpfm13,10
Z1_Command return as byteNo/Char/$Hex; 0/þ/FE 1/=/3D 2/S/53 3/,/2C 4/1/31 5/
/0D
A2_listen return Op _ id , char _ id dec value; S_1_49
B1_POLL command echo follows; þPOLL13,10
Z1_Command return as byteNo/Char/$Hex; 0/þ/FE 1/=/3D 2/N/4E 3/,/2C 4/0/30 5/,/2C 6/0/30 7/
/0D
B1_Poll return Op char _ Handle char _ Id char are N_0_0
B1_POLL command echo follows; þPOLL13,10
Z1_Command return as byteNo/Char/$Hex; 0/þ/FE 1/=/3D 2/N/4E 3/,/2C 4/0/30 5/,/2C 6/0/30 7/
/0D
B1_Poll return Op char _ Handle char _ Id char are N_0_0
```

Returns b = S,id E on success, or b = E,code E on error; see Error Codes. Use the returned listener id with PATH or CLOSE commands.

There are a maximum of four listeners available. Issuing additional LISTEN commands when all four listeners are already established will result in a response of b = E,4 E (NO_FREE_LISTENER) until a CLOSE command is used to free a listener.

- Thereafter it loops through PollEvents, with POLL command with some API snippets;

```
m
A2; þLISTEN:HTTP,/fptm13,10        return in API format          b=S,1
B1; POLL -> POLL return API format  b=N,0,0            no conn cntr 86
B1; POLL -> POLL return API format      b=P,5,1
BMin; P, ID match        B2; þARG:5,txt13
txt string from webpage; þ=S,fghij
þREPLY:5,200,13               B3; REPLY return API format        b=S,0
B1; POLL -> POLL return API format      b=S,5,0
B1; POLL -> POLL return API format  b=N,0,0            no conn cntr 18
B1; POLL -> POLL return API format      b=P,5,1
BMin; P, ID match        B2; þARG:5,txt13
txt string from webpage; þ=S,klmno
þREPLY:5,200,13               B3; REPLY return API format        b=S,0
B1; POLL -> POLL return API format      b=S,5,0
B1; POLL -> POLL return API format  b=N,0,0            no conn cntr 13
B1; POLL -> POLL return API format      b=P,5,1
BMin; P, ID match        B2; þARG:5,txt13
txt string from webpage; þ=S,pqrst
þREPLY:5,200,13               B3; REPLY return API format        b=S,0
B1; POLL -> POLL return API format      b=S,5,0
B1; POLL -> POLL return API format  b=N,0,0            no conn cntr 16
```

Returns b = S,id E on success, or b = E,code E on error; see Error Codes. Use the returned listener id with PATH or CLOSE commands.

There are a maximum of four listeners available. Issuing additional LISTEN commands when all four listeners are already established will result in a response of b = E,4 E (NO_FREE_LISTENER) until a CLOSE command is used to free a listener.

- Thereafter it loops through PollEvents, with POLL command;

b POLL[:filtermask] E
Check for activity like incoming HTTP GET/POST requests, HTTP/WebSocket/TCP connections/disconnections, and incoming WebSocket/TCP data.

b = N:0,0 E

No connection activity has occurred since the last POLL command.

b = G:handle,id E
Received HTTP GET request that matched a listener id's path and is now assigned to the connection handle for processing.
Handle; The connection identifier to use for this request

Id; The identifier of the listener that matched the request

b = P:handle,id E

Received HTTP POST request that matched a listener id's path and is now assigned to the connection handle for processing.

We get P_5_1

This means listener id 1 matches the assigned listener id 1 and handle 5 is now assigned.
Id; The identifier of the listener that matched the request;

- *PATH:handle to get the associated connection path*
- *PATH:id to get the associated listener path (which may end in '*')*
- *Id as a unique identifier of the associated listener; alternative of using PATH:id or PATH:handle and parsing the path*
- *ARG:handle,... to retrieve HTTP POST query/body arguments*
- *RECV:handle,... to get the HTTP body*
- *REPLY:handle,... [+ SEND:handle,...] to respond to POST request*

So we have to use the ARG handle to retrieve the http POST arguments;

We get ARG:5,txt 13

b ***ARG***:handle,name E
Retrieve HTTP GET/POST's name argument (in query or body) on connection
Handle An active connection handle; returned by ***POLL***.

Name The argument name to retrieve

So our handle is still 5 and our Name to retrieve is ; "txt"

The next thing is to receive the txt string from the webpage;

Returns b =S,value E *on success, or* b =E,code E *on error; see Error Codes.*

We get =S,abcde where "abcde" is from the webpage

b =S:handle,0 E ; *A* ***REPLY*** *or* ***SEND*** *operation completed successfully.*

Finally we go to PostReqAck and send REPLY,5,200,13 and get a return string of 0, indicating the operation was completed successfully.

GetReqSrt/ ARG	input value from web	GetReqAck
ARG:5,txt,13	abcde	S_0_48
Handle is 5 from POLL – 5 Name is text E is 13 - 13	This is the value from s,value E	S,0 completed successfully 0D=13(48) is end

3. Wi-Fi connected Laptop load webpage

- Join wx-de09bc on the wireless connection -> connected
- It either requests “additional logon information required”
- Or, Launch Firefox. It defaults to Configuration Page or look up http://esp8266.nonet/index.html
- Make sure text-page-to-micro.html is loaded into wx files at configuration page
- New tab -> http://192.168.4.1/files/text-page-to-micro.html
- Starts out with page . . .Type up to 5 characters, then click send
- In the box type . . . . . abcde
- Click Updatesend button -> To micro : txt = abcde

Setup:

✅ Upload this page to your Wi-Fi module.
✅ Open this page from http://your-wi-fi-module's-ip/files/text-page-to-micro.html.
✅ Run "Text Page to Micro Host.bs2" with your BASIC Stamp 2 module.
📖 More info at http://learn.parallax.com/pbasic-wx-wi-fi.html.

Type up to 5 characters, then click Send:

[ ] Send

Waiting...