SOFTWARE CONTROL SYSTEMS FOR SMART ANTENNA

Aatish Gandotra[1] , Tejaswi Singh[2]

[1]*Student, Electronics and Communication Engineering, Guru Tegh Bahadur Institute of Technology, Delhi, India*
[2]*Student, Computer Science Engineering, Guru Tegh Bahadur Institute of Technology, Delhi, India*

Abstract

A PCB containing microcontroller provides suitable DC voltages to the phase shifters and generates the smart antenna array beam steering. The detected WiFi signals are transferred into a mobile device through a WiFi adapter. This chapter will focus on the software design to automatically control the complete smart antenna array system. Since two microcontrollers PIC18F4550 and LPC1768 are used to build control PCBs. There are also two specifically designed software programs developed in order to configure the individual PCB. For the PIC18F4550, a graphical user interface (GUI) was developed to communicate between a laptop and the control PCB. The GUI sends commands to a Microchip compiler called MPLAB and transfers the control C code into a Hexadecimal (Hex) document. Through the Bootloader program, this Hex code will be copied into the microchip PIC18F4550 and then configures the digital potentiometers to generate variable output voltages. A script using VB is made to link all of the control steps automatically.

Key Words: *Software Control System, Smart Antenna, Manual Control, Switching Control, Automatic Control.*

---***---

1. INTRODUCTION

Several improved versions of the GUI are investigated for the LPC1768 control PCB. By utilizing the microcontroller LPC1768, it is able to achieve the real time control, which means there is no need for the Bootloader program again after initialization. Due to the advantage of LPC1768, two advanced GUIs have been implemented. Both of the GUIs allow the user to detect service set identity (SSID) and received signal strength indicator (RSSI) of WiFi signals surrounding the mobile device. The basic GUI is able to manually configure the digital potentiometers and display the WiFi information. The control signals are transformed from GUI to a virtual COM port, which is established by the FT232RL, and directly go into the microcontroller. By varying the phase shifter, the beam direction of the antenna array could be changed in order to analyze and optimize the signal strength of the received WiFi signals. Moreover, this chapter provides beta version of an advanced software package, which performs a basic smart antenna adaptive beamforming.

There are mainly four sections in this chapter. Section 1.1 presents the software implementation introduction. GUI designs for PIC18F4550 and LPC1768 are demonstrated in Section 1.2 and Section 1.3, respectively. Finally, Section 1.4 summarizes this chapter.

1.1 Software Implementation for P1C18F4550 Control System

Figure 1.1 presents the block diagram of software implementation using PIC18F4550 microcontroller.

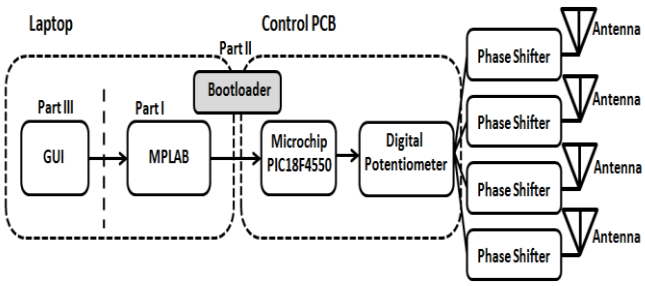


Figure 1.1: Functional Block Diagram for PIC18F4550

The software control system is generally comprised of three parts, as illustrated in Figure 1.1. Part I contains the main algorithm to configure the microchip and digital potentiometers, which simulates the SPI communication. The program is made in C language and compiled in Dev-C++ [2]. Then the C code is transferred into microchip C language to set correct configuration bits, suitable I/O ports and accurate delay time. Finally, MPLAB compiles the program again and generates a hexadecimal document which could be recognized by the microchip. Part II in the software design is the Bootloader program with the function of communicating between laptop and microcontroller PIC18F4550. After initialization (discussed in Chapter 5), the hexadecimal code generated by MPLAB is transferred through Bootloader into the microchip. Finally, Part III is a GUI designed in VC++ for Windows system. The GUI provides two control methods for the phase shifter: manual control and switching control. In the manual control, some specific values will be sent to the digital potentiometers and generate particular DC voltages. While in the switching control, the output voltages are in square waves. This GUI is used for briefly evaluating the digital potentiometers and phase shifters. More advanced control method will be demonstrated using microchip LPC1768.

1.1.1 Programed PIC18F4550

There are three programs in PIC18F4550: SPI communication, manual control and switching control. These C codes have been evaluated in Dev- C++. Dev-C++ is an integrated development environment (IDE) distributed under the GNU general public license for programming in C and C++. It is bundled with MinGW, a free compiler. The IDE is written in Delphi.

1.1.1.1 SPI Communication

A virtual SPI communication has been established between microcontroller PIC18F4550 and digital potentiometer AD5290. The algorithm is based on the potentiometer's timing diagram.

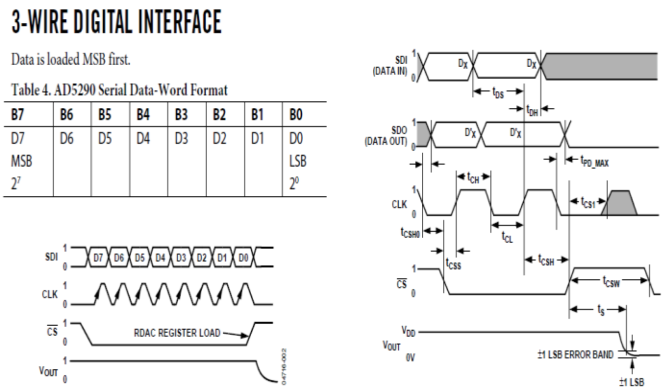


Figure 1.2: Digital Potentiometer AD5290 Timing Diagram [132]

PIC18F4550 controls three signal lines for these digital potentiometers: CLK, and SDI. Clean transitions are needed by the positive edge sensitive CLK input to avoid clocking incorrect data into the serial input register. When is low, the potentiometer loads data into the serial register on each positive clock edge and the MSB of the 8-bit serial is loaded first. The data setup and data hold times determine the valid timing requirements. After eight clock cycles, the 8-bit serial input data register transfer the words to the internal RDAC register and the line returns to logic high. Extra MSB bits will be ignored.

There are two methods to configure the potentiometers: manual control and switching control.

1.1.1.2. Manual Control

In manual control, it is able to control digital potentiometer to generate specific voltage levels to the phase shifter. An example code showing configure only one potentiometer using C in Dev- C++ is as follows:

```
#include <conio.h>
void init();
void write_byte(uchar data);
int CLK, dout[8], cs_bar;
void init()
{        cs_bar=1;
CLK=0;            }
{        uint i,t;
for (i=1;i>0;i--)
{        init();
write_byte(0xfb);//11111011,251
}
for(t=0;t<8;t++)
{  printf("%d",dout[t]); }
getch();
return 0;                }
void write_byte(uchar data)
{        uint i;
uchar temp;
cs_bar=0;
for (i=0;i<8;i++)
{
 CLK=0;
temp=data&0x80;
if (temp==0x80)
dout[i]=1;
else
dout[i]=0;
CLK=1;
}
data=data<<1;    }
cs_bar=1;  }
```

This program is exactly following the timing diagram provided in Figure 1.2. The purpose of this code is to write an 8-bit data into the microcontroller and then transfer the data into a digital potentiometer bit by bit on positive clock cycle. The delay time is ignored to simplify the program. If a number 251 is written into the microchip PIC18F4550, its output to the digital potentiometer should be "11111011". A Printf function is added into the C code, so the result of 1111011 is showing on the screen, as presented in Figure 1.3.

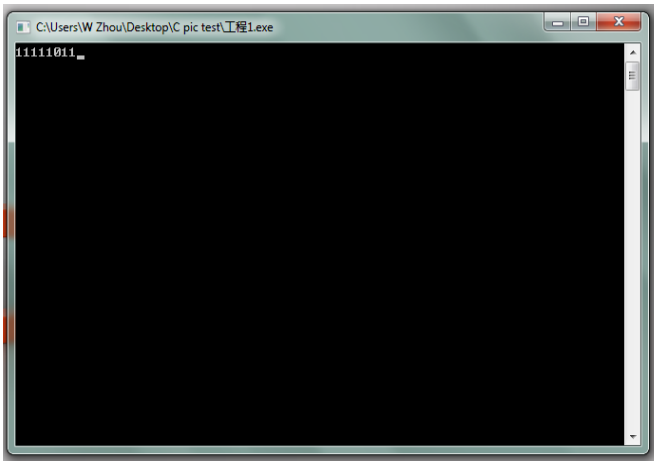


Figure 1.3: C Code Result for SPI Communication

From the results in Figure 1.3, the virtual SPI communication has been successfully established between PIC18F4550 and the digital potentiometers. The algorithm is fully matching the timing diagram. Then, the C code is re-written in MPLAB to configure the microchip I/O ports. MPLAB integrated development environment (IDE) is an integrated toolset for development of embedded applications employing Microchip's PIC and dsPIC microcontrollers [1].

The following example demonstrates using PIC18F4550 configure only one potentiometer in MPLAB.

```
#include<p18f4550.h>
// Include the 18f4550 head file
#define REMAPPED_RESET_VECTOR_ADDRESS
0x1000 // Memory from 0x1000
#define cs_bar LATCbits.LATC1   // Set PIN RC1 as
#define dout LATAbits.LATA0     // Set PIN RA0 as output
data
#define CLK LATCbits.LATC2      // Set PIN RC2 as CLK
extern void _startup (void);    // Set start up
#pragma code REMAPPED_RESET_VECTOR =
REMAPPED_RESET_VECTOR_ADDRESS
void _reset (void)
{       _asm goto _startup _endasm      }
#pragma code                    // Set the suitable
configuration bits
void delay(uint x);
void init();
void write_byte(uchar data);
void delay(uint x)              // Make a delay function
{       uint a,b;
for(a=x;a>0;a--)
for(b=275;b>0;b--); }   // After simulation, x represent x
milliseconds
void init()
{       TRISAbits.TRISA0=0;     // Set PIN RA0 as an
output, for the output data TRISCbits.TRISC1=0;     // Set
PIN RC1 as an output, for TRISCbits.TRISC2=0;      // Set
PIN RC2 as an output, for CLK
cs_bar=1;               // Before data transfers,     should be
logic high


CLK=0;  }               // Initialise the clock signal
void main()
{       uint i;
for (i=1;i>0;i--)       // Run the main function once
init();
write_byte(0x84);//10000100,132    }//Write 132 into the
microcontroller's memory
void write_byte(uchar data)
{       uint i;
uchar temp;
cs_bar=0;               // Data starts to transfer,     should be
logic low
Nop();                  // Make a delay for
for (i=8;i>0;i--)       // 8-bit data needs the transfer
function 8 times
{       CLK=0;          // CLK goes to low to simulate a
clock signal
temp=data&0x80;   // Check this bit is 0 or 1
if (temp==0x80)   // If this bit is 1
dout=1;   // The output data should be set as logic high
else              // If this bit is 0
dout=0;   // The output data should be set as logic low
Nop();            // Wait for the data processing
CLK=1;            // CLK returns to high to finish a clock
cycle
Nop();            // Wait for the clock signal
data=data<<1;   } // Left shift the data to write next bit into
the memory
CLK=0;                  // After data transfers, CLK
returns to 0
cs_bar=1;               //After the function runs for
eight times,
// all of the eight bits have been
// transferred into the memory,
}       //      returns to high and finish the write_byte
function
```

Some I/O ports of PIC18F4550 are utilized as , CLK, and SDI signals to simulate the SPI communication. In the code, data 132 is transferred into the digital potentiometer.

In MPLAB, after compiling, the code is translated into a hexadecimal document. Bootloader program is used to send the ".hex" file into PIC18F4550. After resetting the control PCB, a voltage of 15.47V is measured at the terminal W of the digital potentiometer, which could configure the phase shifters in the smart antenna array.

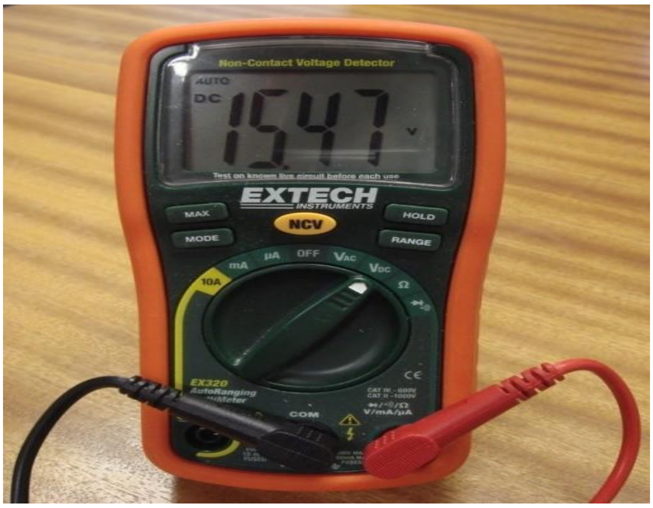


Figure 1.4: Measurement Result of PIC18F4550 Manual Control

Figure 1.4 illustrates the measurement of PIC18F4550 manual control for single digital potentiometer, which matches the calculation. Then the algorithm is extended to configure all of the sixteen digital potentiometers together on the control PCB. Finally the PIC18F4550 manual control program will work with the manual control GUI to control the smart antenna array beam steering.

1.1.1.2. Switching Control

In switching control, the target is to generate square waveforms from the digital potentiometers, as presented in Figure 1.5.

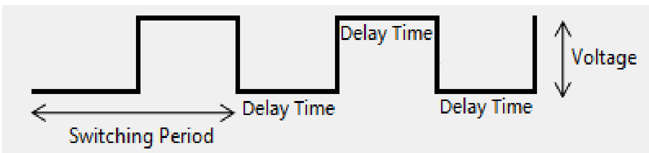


Figure 1.5: Square Waveforms from Switching Control

The fundamental algorithm is similar to the manual control, however, in this implementation, the delay time is very important. In the microchip C code, a delay time has been added after sending the 8-bit values and a for-loop is included in the main function to judge the delay time.

```
void delay(uint x)
{        uint a,b;
for(a=x;a>0;a--)
for(b=275;b>0;b--);}
```

The value of b determines the accuracy of delay time, which has been simulated in MPLAB, as illustrated in Figure 1.6.

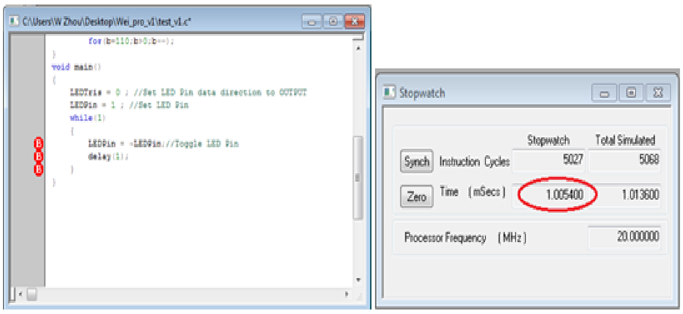


Figure 1.6: Delay Time Simulation in MPLAB

In Figure 1.6, several breakpoints (Red B) are set at the beginning and end of the delay function. The Stopwatch in MPLAB is used to observe the running time. After simulations and optimizations, the value of b is set at 275 and x equals to 1. The program will run for 1 millisecond (Red cycle), and later by controlling the value of x, the required milliseconds can be achieved.

The simulated delay time function has been added into the switching control main code and transferred into the microcontroller PIC18F4550. Figure 1.7 demonstrates the measured results by an oscilloscope. In this code, the DC voltage is switching between 0 and 12V with a delay time of 9 milliseconds.

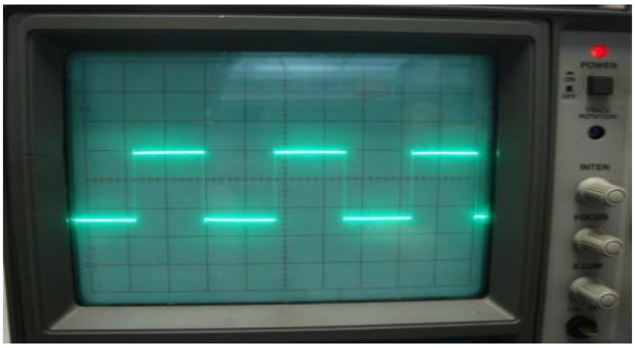


Figure 1.7: Measured Result of the Switching Control

1.1.2 Graphical User Interface for PIC18F4550

A graphical user interface is built to automatically configure the PIC18F4550 control PCB.

Figure 1.8 shows the communication loop between GUI and microchip.

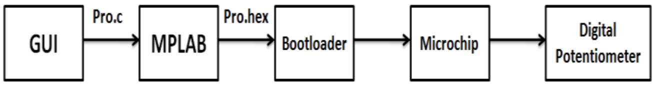


Figure 1.8: Control Loop between GUI and Microchip

The control commands from GUI will automatically generate a C code. MPLAB transfers the C code into a hexadecimal document which will be sent into the microchip through Bootloader. Finally the microcontroller configures the digital potentiometers to provide suitable DC voltages to the phase shifter in smart antenna array.

Figure 1.9 shows the GUI main interface for PIC18F4550. Two control methods are provided according to the algorithm described in Section 1.2.1.

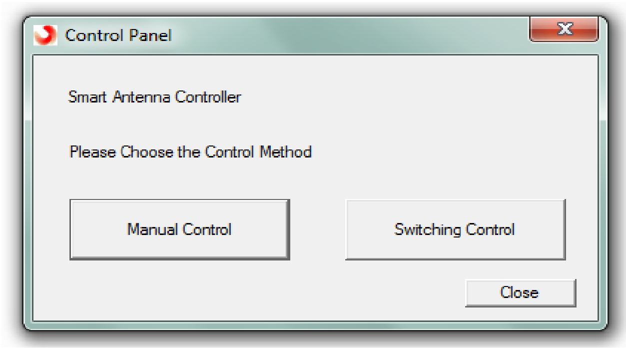


Figure 1.9: GUI Main Window for PIC18F4550

1.1.2.1 Manual Control

The GUI is made using Microsoft Visual Studio 2010 [138], which is a commercial integrated development environment (IDE) product engineered by Microsoft for the C, C++, and C++/CLI programming languages. In this software program, the main section is written in C++.

The manual control interface is illustrated in Figure 1.10.

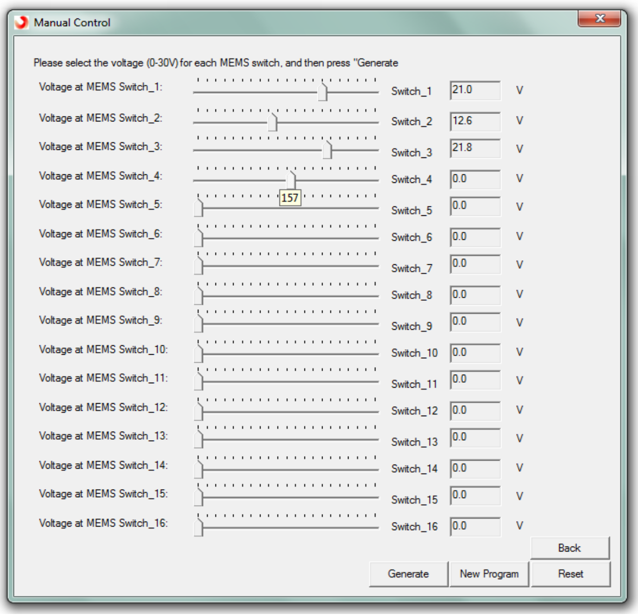


Figure 1.10: Manual Control GUI for PIC18F4550

In manual control GUI, there are sixteen sliders to control the digital potentiometers, with the voltage level from 0V to 30V in the precision of 0.1V. The scale is above the slider and there stands a number showing the position. By using the scales and sliders, it is able to configure the digital potentiometers quickly and accurately. The text boxes are on the right demonstrating the output voltages. The required voltage of the phase shifter is from 0V to 13V, which can be fully covered by this implementation. After pressing the Generate button, a script written in VB will send these sixteen voltage values into a predefined C code. Then MPLAB reads this C code, compiles the program and transfers it into a hexadecimal file. Finally, Bootloader burns the hex document into the microcontroller PIC18F4550 and control the digital potentiometers. By using the manual control GUI, it is able to manually configure the phase shifters in the smart antenna array and estimate the beam steering.

1.1.2.2 Switching Control

In switching control, the digital potentiometers will generate square waveforms to the phase shifters and the main beam of the smart antenna will switch just between two directions. It is a basic evaluating method for the smart antenna array.

Figure 1.11 shows the output waveforms. The voltage levels are different but they share an identical switching period. The software interface is displayed in Figure 1.12.

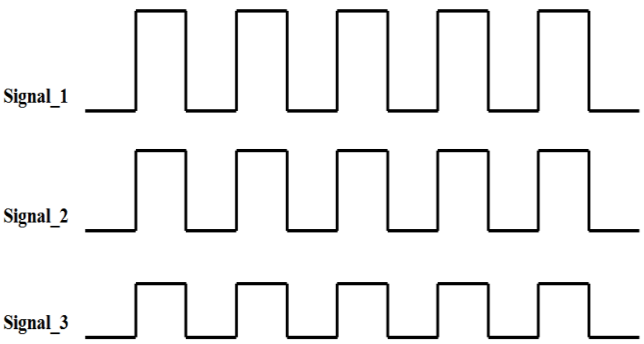


Figure 1.11: Output Square Waveforms

Same as the manual control, for the square waveform, the voltage range is from 0V to 30V with a precision of 0.1V. There are scales and positions of the sliders added into the GUI to achieve the accurate control. Furthermore, in this switching control application, the delay time has been added into the design to setup a switching period. Also, a script is made to automatically link the GUI to MPLAB, Bootloader and PIC18F4550.

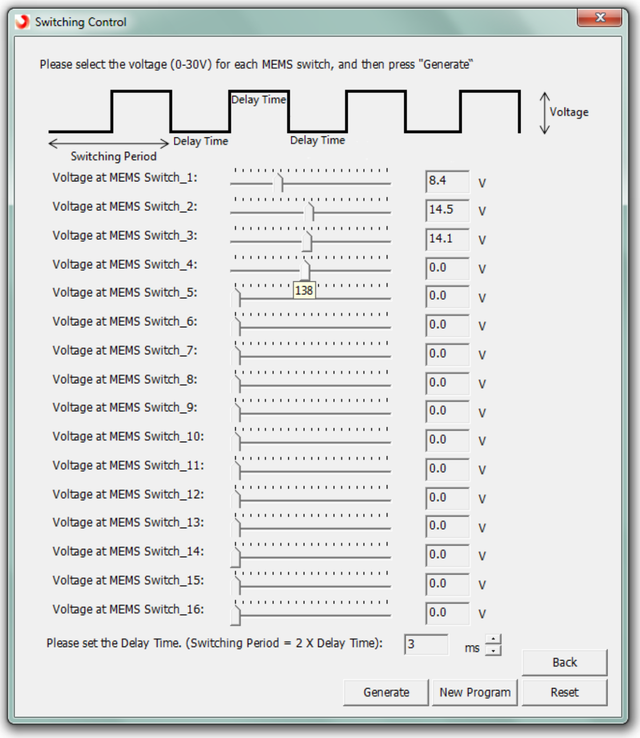


Figure 1.12: Switching Control GUI for PIC18F4550

Based on research and investigation, it is noted that PIC18F4550 is difficult to achieve the real time control, which means the Bootloader program cannot be ignored during the control loop. The script built in the GUI is able to generate an automatic control but each command requires a long data transfer period and the program is not stable. The basic control methods (manual control and switching control) have already made the algorithms complicated. So the GUI of PIC18F4550 is only used to generally estimate the digital potentiometers and the phase shifters. More advanced software programs are developed using the microcontroller LPC1768.

1.2 Software Implementation for LPC1768 Control System

1.2.1 Programme LPC1768

Figure 1.13 shows the block diagram for programming LPC1768. The microchip control code is written in C language and compiled by an online compiler provided by mbed, as illustrated in Figure 1.14. The compiler generates a binary document. A DOS based program transfers the binary file into a hexadecimal code. Flash Magic burns the .hex file into the microcontroller LPC1768, as demonstrated in Figure 1.15 and Figure 1.16. Flash Magic is a PC tool for programming flash based microcontrollers from NXP utilizing a serial protocol [3]. An ISP programmer is communicating between laptop and control PCB.

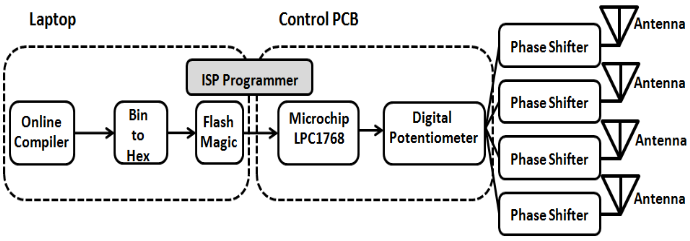


Figure 1.13: LPC1768 Programming Block Diagram

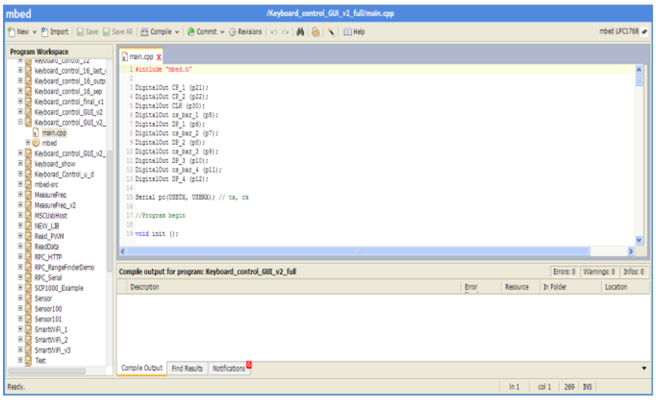


Figure 1.14: LPC1768 Online Compiler

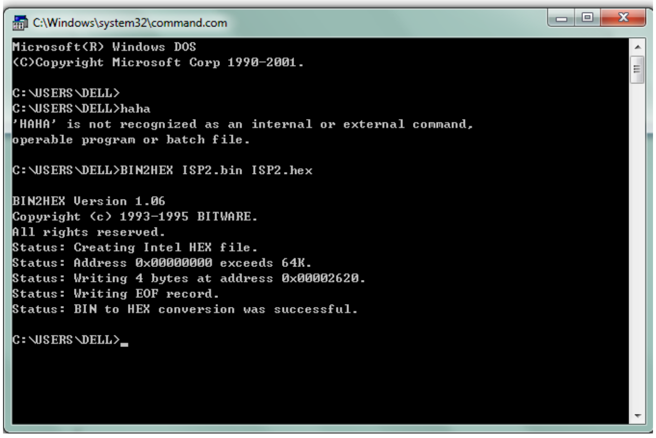


Figure 1.15: LPC1768 BIN to HEX Transformer

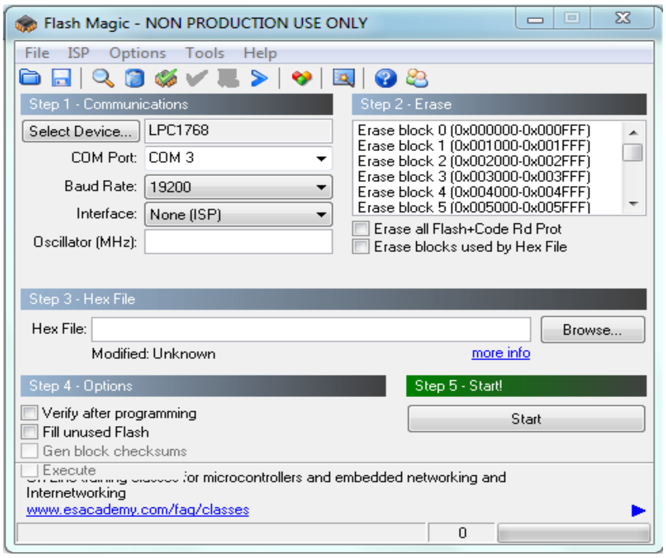


Figure 1.16: Flash Magic to Program LPC1768

Compared to PIC18F4550, the microchip LPC1768 is more advanced. After initializing the microcontroller, as shown in Figure 1.13, the compiler, transformer, flash magic and ISP programmer are not required in the final communication. When the configuration bits and algorithms have been setup correctly, the LPC1768 will generate a virtual COM port (by the FT232RL chip), which can be recognized by a terminal emulator named Tera Term [140]. The GUI for LCP1768 can directly control the digital potentiometers through Tera Term. It is able to achieve the real time control. The communication block diagram for LPC1768 is presented in Figure 1.17.

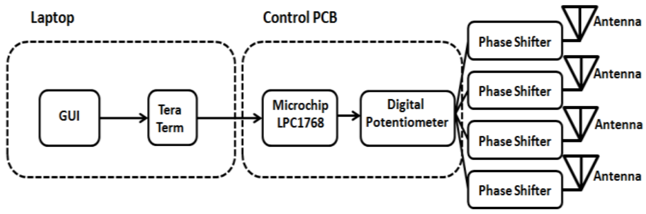


Figure 1.17: Real Time LPC1768 Communication Block Diagram

Similar to PIC18F4550, the LPC1768 also controls the digital potentiometers using virtual SPI communication, and with the same algorithms in Section 1.2.1.

1.2.2 Graphical User Interface for LPC1768

There are also two GUI developed for LPC1768: Manual Control and Automatic control. Both of the software programs are able to detect service set identity (SSID) and received signal strength indicator (RSSI) of WiFi signals surrounding the laptop.

The GUI software can be installed on a computer with the following minimum requirements:

- PC Compatible with Windows XP/7/Vista/ 8 – 32 or 64 bit
- At least 2 free USB Ports

Usually Windows 7 and later versions will recognize the LPC1768 microcontroller automatically and there will be no need to install the drivers. However, for previous Windows version, two drivers for FT232RL microchip and the WiFi adapter are required to be installed.

1.2.2.1 Driver Installation

The driver for LPC1768 has been written into FT232RL microchip and the program will automatic start when the USB port is connected to a PC running Windows. Figure 1.18 presents the steps to install the FT232RL.

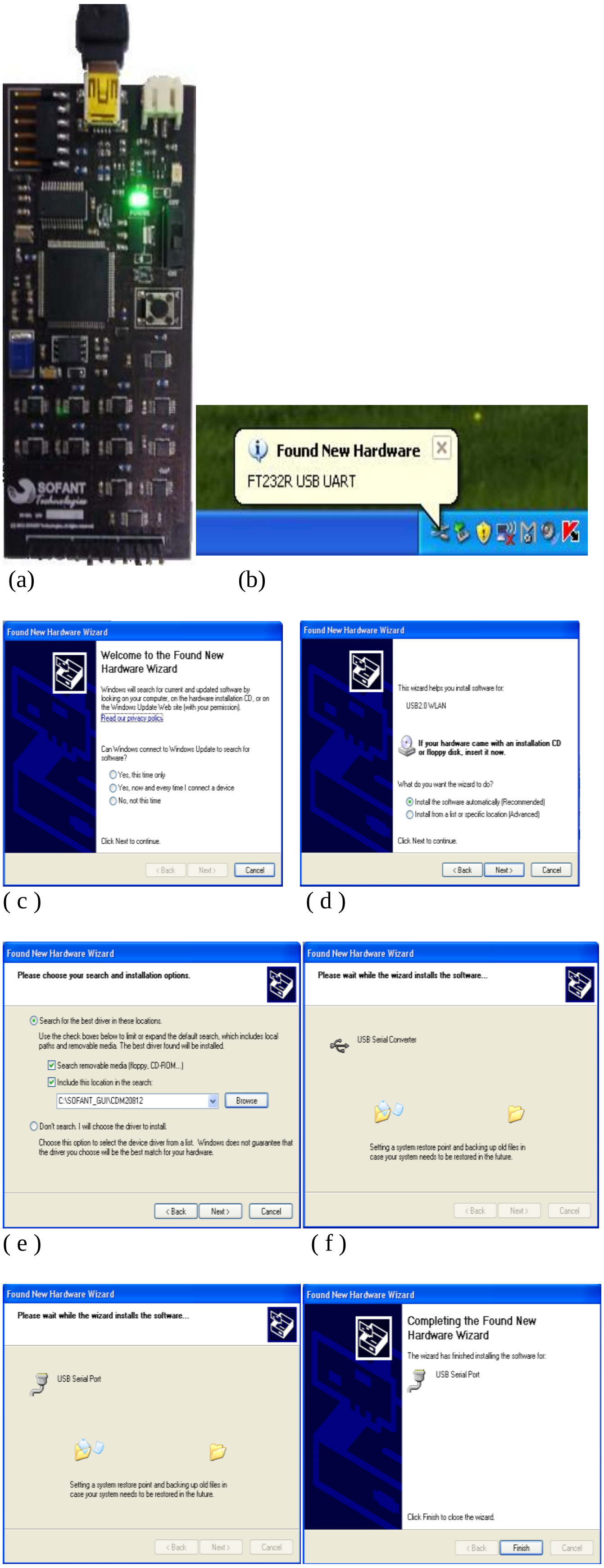


Figure 1.18: FT232RL Diver Installation

FT232RL provides a USB serial port which can be recognized by Windows system. The laptop communicates to LPC1768 through FT232RL USB to UART.

The installation steps for WiFi adapter are illustrated in Figure 1.19.

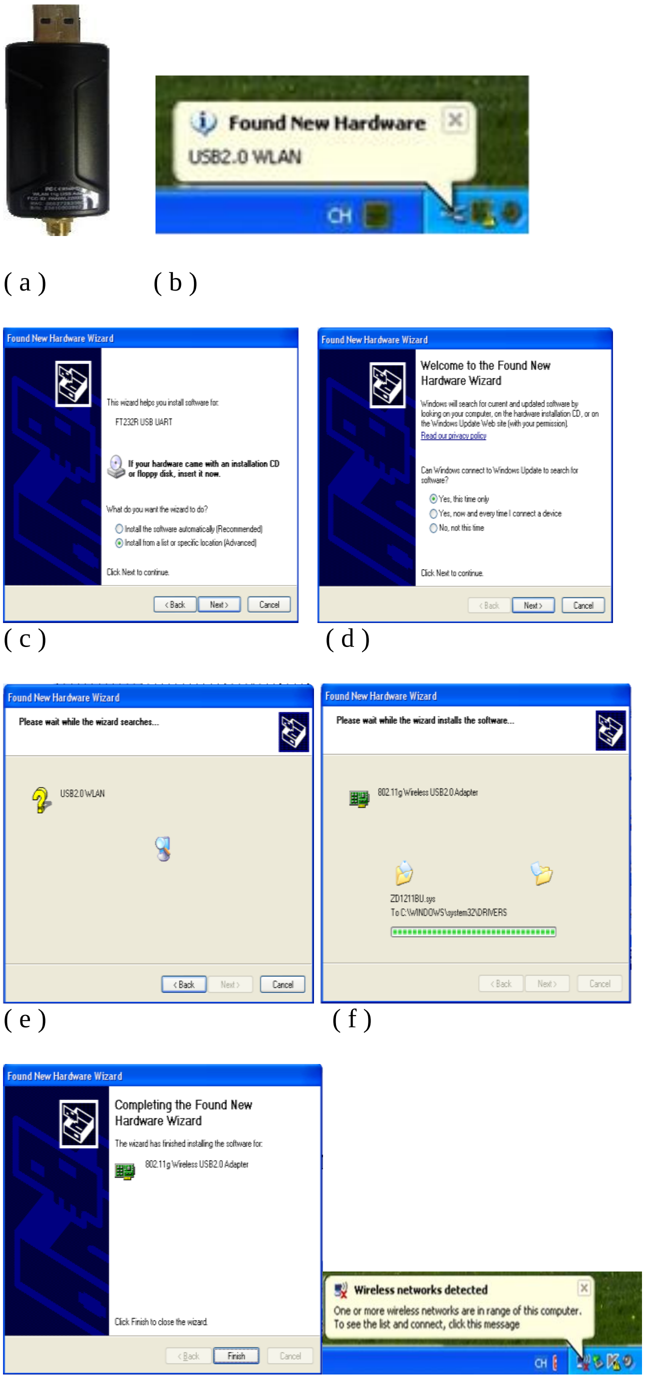
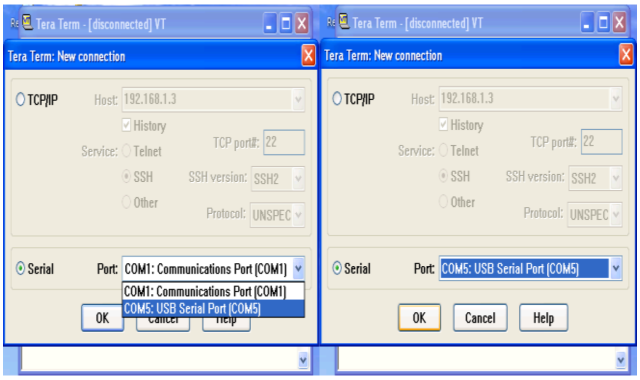


Figure 1.19: WiFi Adapter Driver Installation

After installation, the WiFi adapter will replace the laptop's original WiFi antenna and the smart antenna array will be used to detect WiFi signals around the laptop. The following sections will focus on the GUI design to configure the smart antenna array beam steering.

1.2.2.2 Manual Control

As discussed in Section 1.3.1, the GUI sends the control commands into Tera Term and then directly transfers to the microprocessor LPC1768. Figure 1.20 presents the Tera Term software which makes LPC1768 a virtual COM port to the laptop.



(a) (b)

Figure 1.20: Tera Term Connection Setup

Figure 1.21: Manual Control GUI Start Screen for LPC1768

Figure 1.21 presents the GUI start screen developed for microprocessor LPC1768. This software program is written in VC++ and C#. After clicking the Basic SmartWiFi button, the main interface will appear, as displayed in Figure 1.22.

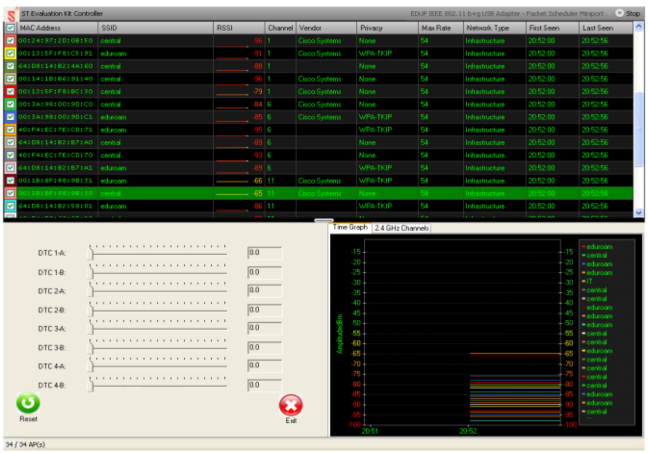


Figure 1.22: Manual Control GUI Main Screen for LPC1768

In the upper right corner, there is a drop down menu item to select the wireless adapter for evaluation. In the smart antenna array system, an EDUP wireless adapter is used in the implementation. So in the menu, "**EDUP IEEE 802.11 b+g USB Adapter – Packet Scheduler Miniport**" is selected.

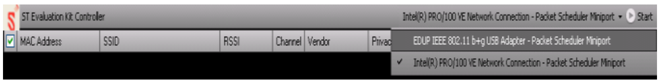
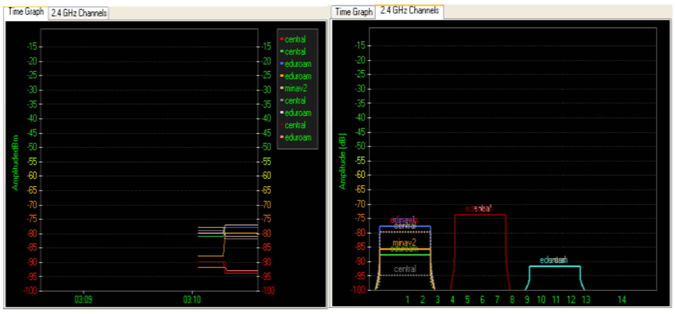


Figure 1.23: Wireless Adapter Selection for LPC1768

There are generally three sections in the software design. On top is the main window showing MAC Address, SSID, RSSI, Channel, Vendor, Privacy, Max Rate, Network Type and Detected Time for all of the WiFi signals around a laptop, as demonstrated in Figure 1.24.

Figure 1.24: WiFi Information of the GUI for LPC1768

On the bottom right, a window is drawing all of the real time signal strength curves (in dBm) for the WiFi signals (as shown in Figure 1.25(a)). Another tab in this window illustrates the signal amplitudes in different channels.



(a) (b)

Figure 1.25: (a) WiFi Signal Strength Real Time Curve (b) WiFi Signal Strength in Different Channels

On the bottom left, there exist eight real time sliders controlling the eight Hittite analogue phase shifters on the smart antenna board.

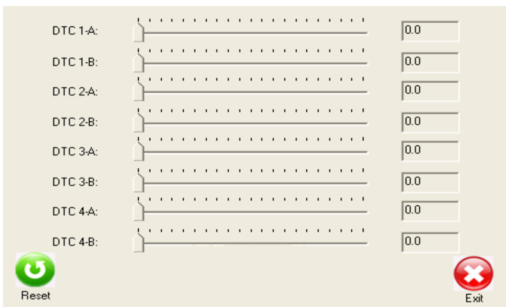


Figure 1.26: Real Time Control Slider for LPC1768

By changing the voltages to the phase shifters, the main beam direction of the antenna array is rotated, which generates a variation of the received RSSI in the manual control GUI. Figure 1.27 shows the evaluated results. Different voltage levels are provided to the antenna array and the received WiFi signal strength curves are able to reflect the diversity.

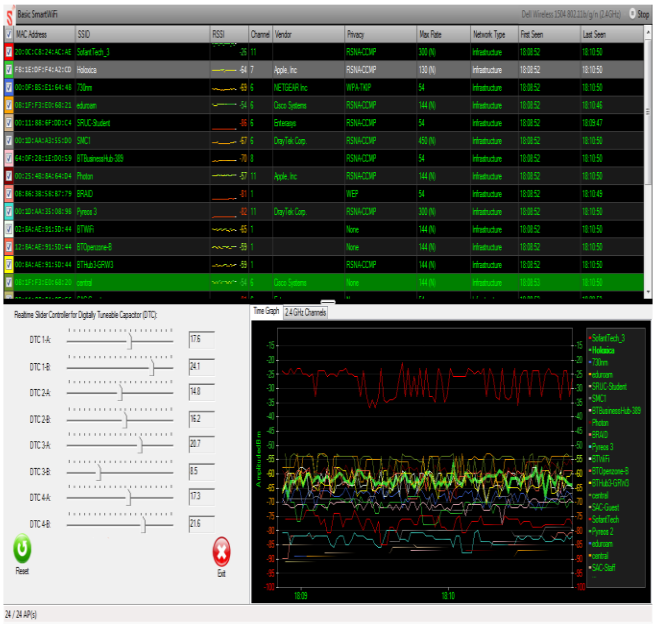


Figure 1.27: Manual Control Evaluating Results for LPC1768

1.2.2.3 Automatic Control

An improved version of the GUI has been developed which can provide the smart antenna array automatic control. The main screen is illustrated in Figure 1.28.

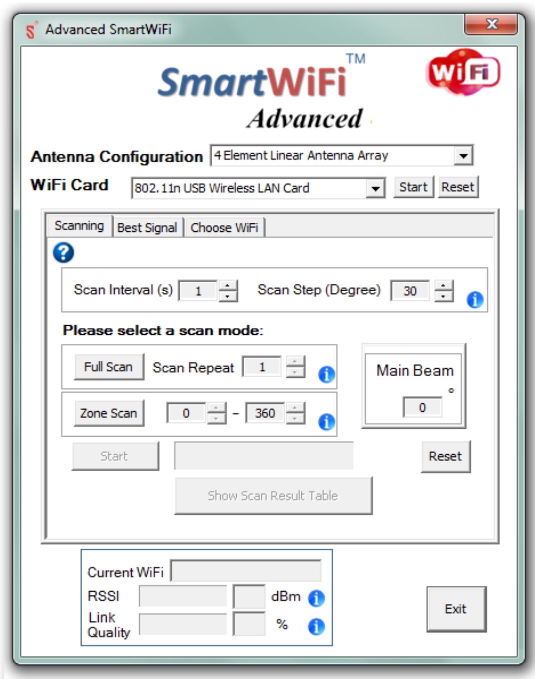


Figure 1.28: Automatic Control GUI for LPC1768

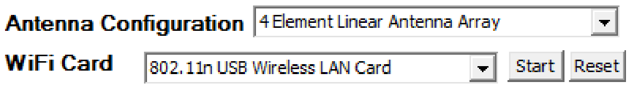


Figure 1.29: Antenna Array and WiFi Adapter Selection

In the current smart antenna array implementation, the four-element linear antenna array with Archimedean spiral slots is used. In Figure 1.29, the “4 Element Linear Antenna Array” and “EDUP 802.11 b+g USB adapter” should be selected. This automatic GUI is also compatible with more elements antenna arrays. By selecting the EDUP USB adapter, the original WiFi card of the laptop will be disabled. The smart antenna array will detect and monitor the WiFi signals for the control laptop.

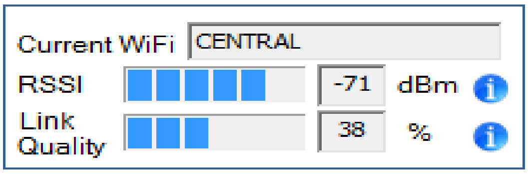


Figure 1.30: Current Connected WiFi Information

On the bottom left of the advanced GUI, the current WiFi information is displayed, as shown in Figure 1.30. Furthermore, in the software program, some information windows help to describe RSSI and Link Quality, as illustrated in Figure 1.31 and Figure 1.32, respectively.

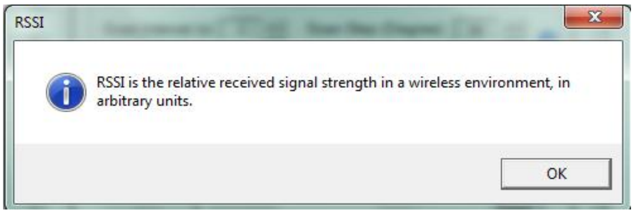


Figure 1.31: RSSI Description

RSSI is the relative received signal strength in a wireless environment, in arbitrary units.

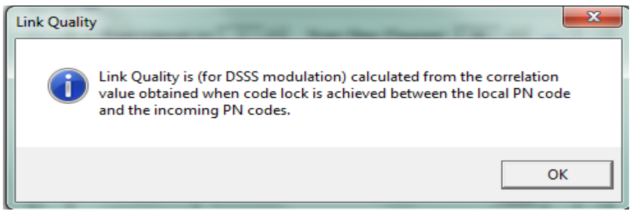


Figure 1.32: Link Quality Description

Link Quality is calculated from the correlation value obtained when code lock is achieved between the local PN code and the incoming PN codes.
There are three functions included in the Advanced GUI for LPC1768: Scanning, Best Signal and Choose WiFi.

1.2.2.3.1 Scanning Function

Scanning: In scanning mode, the program enables scanning of the surrounding environmental for WiFi signals. This takes place by steering the main beam of the smart antenna array

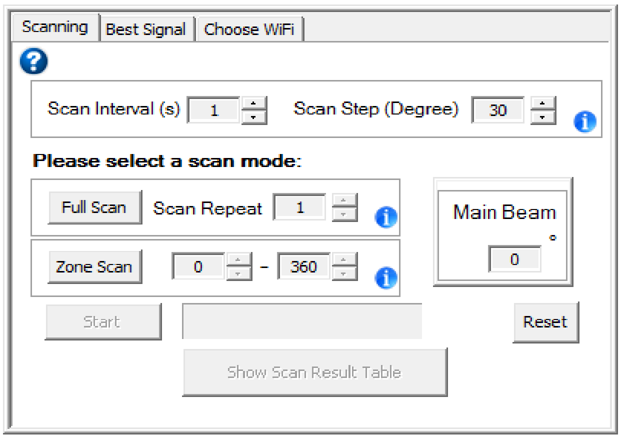


Figure 1.33: Scanning Function of the GUI for LPC1768

In Figure 1.33, the scan interval and scan step can be selected. Scan interval is the time between two sequential scans. Scan step is the angle separation between two sequential scans after steering the beam.

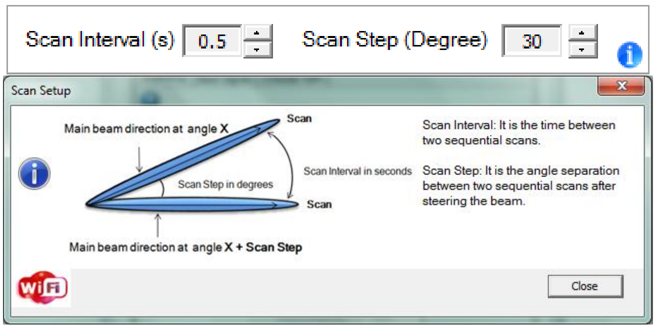


Figure 1.34: Scan Interval and Scan Step Definition

Two modes for the scanning are provided: Full scan and Zone scan.

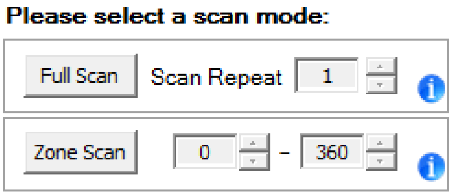


Figure 1.35: Scan Mode Selection

In Full Scan, this mode performs a full 360 degrees scanning for all WiFi signals. The Scan Repeat is a parameter to apply the rescanning for another round, up to 5 times, default value is 1.
After clicking the Start button, the full scan will begin and a bar shows the progress. While scanning, another window near the "Reset" button is showing the current main beam direction, as shown in Figure 1.36.

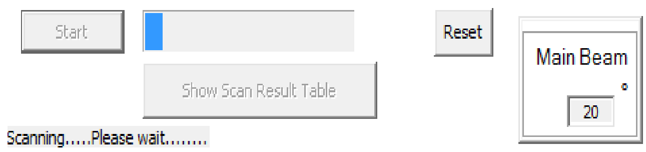
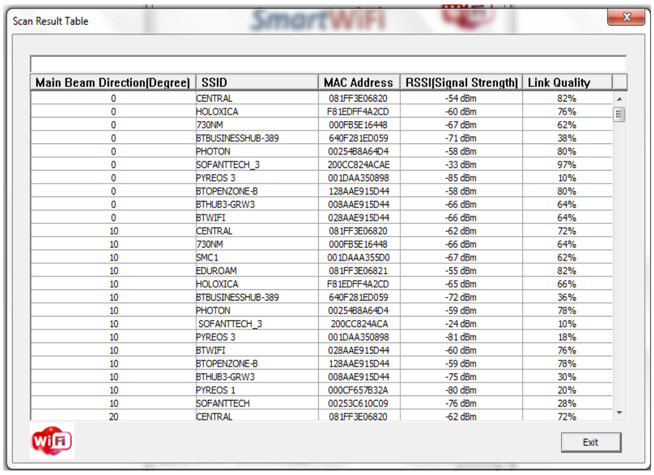


Figure 1.36: Status Bar and Current Main Beam Direction

When the scan finished, a popup table will show the WiFi signals information in surrounding environment. For the current four element linear antenna array, the main beam is able to steer from -50° to +50° for WiFi application. The full scan function has been modified to show only ±50° directions.


Scan Result Table

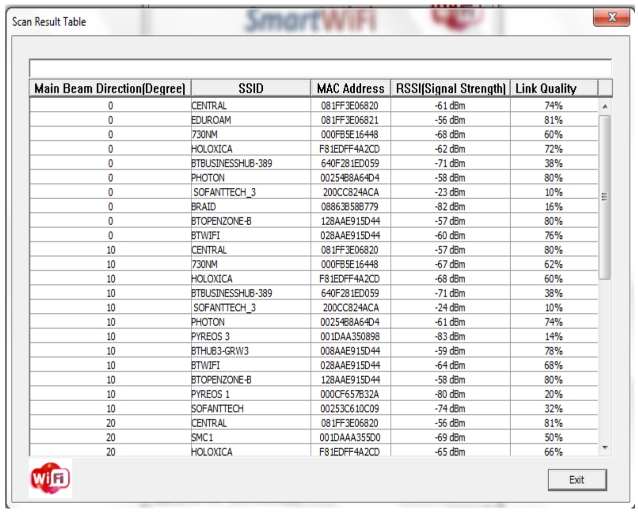
Main Beam Direction(Degree)	SSID	MAC Address	RSSI(Signal Strength)	Link Quality
0	CENTRAL	081FF3E06820	-54 dBm	82%
0	HOLOXICA	F81EDFF4A2CD	-60 dBm	76%
0	730NM	000FB5E16448	-67 dBm	62%
0	BTBUSINESSHUB-389	640F281ED059	-71 dBm	38%
0	PHOTON	00254B8A64D4	-58 dBm	80%
0	SOFANTTECH_3	200CC824ACAE	-33 dBm	97%
0	PYREOS 3	001DAA350898	-85 dBm	10%
0	BTOPENZONE-B	128AAE915D44	-58 dBm	80%
0	BTHUB3-GRW3	008AAE915D44	-66 dBm	64%
0	BTWIFI	028AAE915D44	-66 dBm	64%
10	CENTRAL	081FF3E06820	-62 dBm	72%
10	730NM	000FB5E16448	-66 dBm	64%
10	SMC1	001DAAA355D0	-67 dBm	62%
10	EDUROAM	081FF3E06821	-55 dBm	82%
10	HOLOXICA	F81EDFF4A2CD	-65 dBm	66%
10	BTBUSINESSHUB-389	640F281ED059	-72 dBm	36%
10	PHOTON	00254B8A64D4	-59 dBm	78%
10	SOFANTTECH_3	200CC824ACA	-24 dBm	10%
10	PYREOS 3	001DAA350898	-81 dBm	18%
10	BTWIFI	028AAE915D44	-60 dBm	76%
10	BTOPENZONE-B	128AAE915D44	-59 dBm	78%
10	BTHUB3-GRW3	008AAE915D44	-75 dBm	30%
10	PYREOS 1	000CF657B32A	-80 dBm	20%
10	SOFANTTECH	00253C610C09	-76 dBm	28%
20	CENTRAL	081FF3E06820	-62 dBm	72%

Exit

Figure 1.37: Full Scan Results Using Four-Element Linear Antenna Array

The table in Figure 1.37 presents the full scan results for the four-element linear antenna array with the main beam steering from -50° to +50° in the step of 10°. At each direction, the table is displaying Main Beam Direction, SSID, MAC Address, RSSI and Link Quality. It is clear that along with the beam rotating, the RSSI and link quality are showing different values.

In Zone Scan mode, it allows scanning a defined area or zone by only steering the beam within this defined zone. When the scan finishes, a popup table will show the WiFi signals information within the selected area. The table presented in Figure 1.38 shows the defined zone scan results for the four-element linear antenna array with the main beam steering from -
30° to +30° in the step of 10°.


Scan Result Table

Main Beam Direction(Degree)	SSID	MAC Address	RSSI(Signal Strength)	Link Quality
0	CENTRAL	081FF3E06820	-61 dBm	74%
0	EDUROAM	081FF3E06821	-56 dBm	81%
0	730NM	000FB5E16448	-68 dBm	60%
0	HOLOXICA	F81EDFF4A2CD	-62 dBm	72%
0	BTBUSINESSHUB-389	640F281ED059	-71 dBm	38%
0	PHOTON	00254B8A64D4	-58 dBm	80%
0	SOFANTTECH_3	200CC824ACA	-23 dBm	10%
0	BRAID	08863B58B779	-82 dBm	16%
0	BTOPENZONE-B	128AAE915D44	-57 dBm	80%
0	BTWIFI	028AAE915D44	-60 dBm	76%
10	CENTRAL	081FF3E06820	-57 dBm	80%
10	730NM	000FB5E16448	-67 dBm	62%
10	HOLOXICA	F81EDFF4A2CD	-68 dBm	60%
10	BTBUSINESSHUB-389	640F281ED059	-71 dBm	38%
10	SOFANTTECH_3	200CC824ACA	-24 dBm	10%
10	PHOTON	00254B8A64D4	-61 dBm	74%
10	PYREOS 3	001DAA350898	-83 dBm	14%
10	BTHUB3-GRW3	008AAE915D44	-59 dBm	78%
10	BTWIFI	028AAE915D44	-64 dBm	68%
10	BTOPENZONE-B	128AAE915D44	-58 dBm	80%
10	PYREOS 1	000CF657B32A	-80 dBm	20%
10	SOFANTTECH	00253C610C09	-74 dBm	32%
20	CENTRAL	081FF3E06820	-56 dBm	81%
20	SMC1	001DAAA355D0	-69 dBm	50%
20	HOLOXICA	F81EDFF4A2CD	-65 dBm	66%

Exit

Figure 1.38: Zone Scan Results Using Four-Element Linear Antenna Array

1.2.2.3.2 Best Signal Function

The second tab in the GUI is called Best WiFi, as demonstrated in Figure 1.39. This function enables the user to identify the best available WiFi signals at the location. This is carrying out by fully steering the main beam to enable the full scan the surround environment for the best WiFi signals.

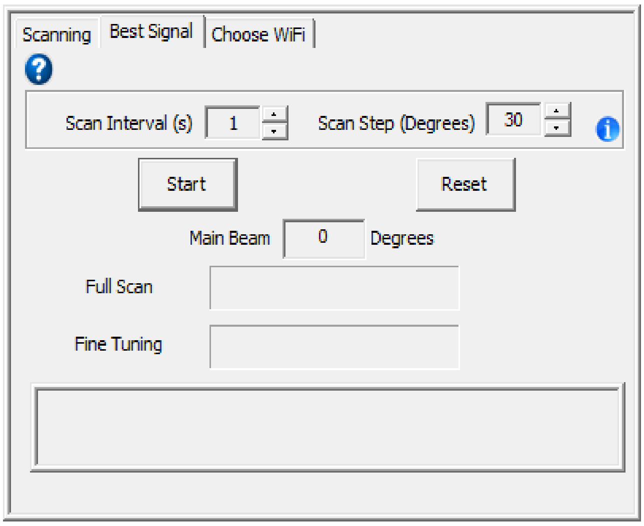


Figure 1.39: Best Signal Function of the GUI for LPC1768

After selecting the scan interval and scan step, the program will automatically perform a full scan and accurate fine tuning. Finally, the detailed best WiFi connect information will be listed, which includes SSID, MAC Address, Signal Strength and Link Quality.

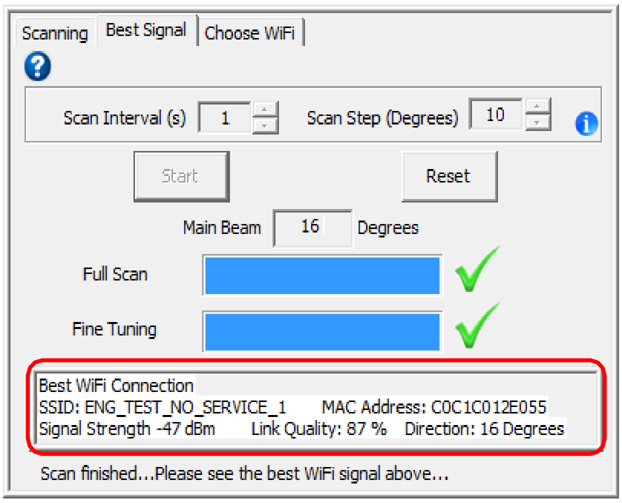


Figure 1.40: Best Signal Scan Results Using Four-Element Linear Antenna Array

1.2.2.3.2 Choose WiFi Function

In this function, the system allows the user to connect to the desired WiFi network. The system will keep steering the beam in order to keep the signal strength above the signal threshold. In this mode, the user has to select the desired WiFi network resulted from the pre- scanning, define the first and second threshold and then start the scanning. The system will automatically connect to the desired wireless network while steering the main beam in the surround environment in order to identify the direction of the best signal strength. The function continues monitoring the performance of the connected signal. If the signal drops below the first threshold, this will trigger the fine tuning mode. If the system fails to recover it or the signal drop below the second threshold, it will switch to the full scan mode. The system will try to recover the WiFi network of the best signal at all times. If the WiFi network is unavailable, the system will display a warning message.

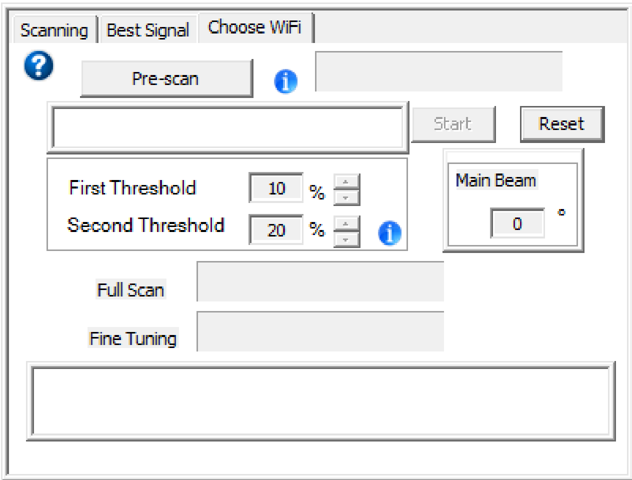


Figure 1.41: Choose WiFi Function of the GUI for LPC1768

Figure 1.41 shows the Pre-scan. In this mode, the system is capturing all WiFi networks (SSID) available to allow the user to select the desired network.

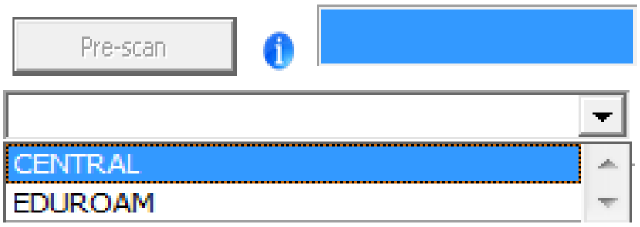


Figure 1.42: Pre-Scan in Choose WiFi Function

The user is able to select a particular WiFi network from the drop down list.

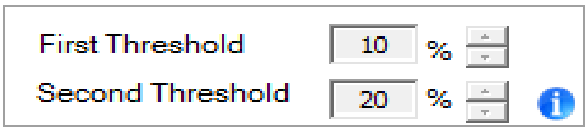


Figure 1.43: Threshold in Choose WiFi Function

Furthermore, it is able to setup the threshold values for the chosen WiFi. In Figure 1.43, First Threshold will trigger the fine tuning to take place in order to keep the signal strength above the first threshold. Second Threshold will trigger the full scan mode to search for the direction with the highest signal strength for the chosen network. The first threshold value is smaller than the second one. After setting up the parameters, the software program automatically performs a full scan and accurate fine tuning. Finally, the detailed chosen WiFi information will be listed, which includes SSID, MAC Address, Signal Strength and Link Quality. The system continues monitoring the RSSI using threshold values to maintaining the strongest WiFi signal.

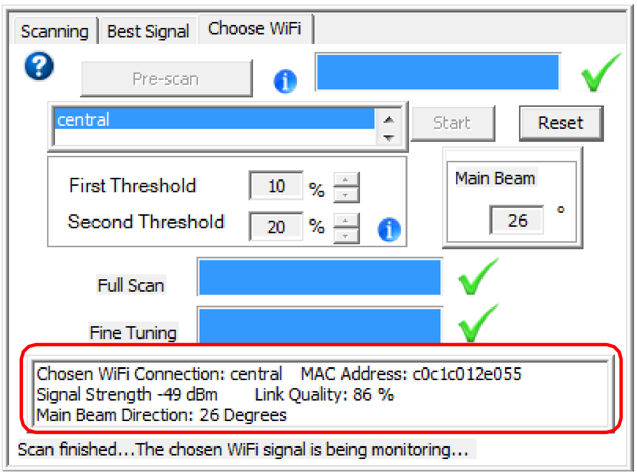


Figure 1.44: Choose WiFi Scan Results Using Four-Element Linear Antenna Array

In the above Scanning, Best WiFi and Choose WiFi functions, all of the algorithms are able to cover the main beam steering from 0º to 360º. However, limited by the four elements linear antenna array characterization, only -50º to 50º beam rotating is performed. Further research will produce a wider scanning range smart antenna array to generate more accurate scanning results.

2. RESULTS

This paper presents the software implementation to automatically configure the smart antenna array system. Two series of control systems are developed for microprocessor PIC18F4550 and LPC1768, respectively.

For microcontroller PIC18F4550, research begins with a virtual SPI communication algorithm between the microchip and digital potentiometers. A graphical user interface has been proposed to control the microchip through a laptop. After initializing PIC18F4550, the GUI transfers the control signal into a Microchip compiler called MPLAB and changes the C code into a Hexadecimal (Hex) document. By utilizing the Bootloader program, this Hex code will be delivered into the microchip PIC18F4550 and then configures the phase shifters on the smart antenna array. Two control methods: Manual Control and Switching Control are provided to achieve particular DC voltage or square waveforms from the digital potentiometers. The PIC18F4550 software control system is able to generally configure the smart antenna array but requires other application to check the WiFi signal variations.

Two improved versions of the GUI are developed for microcontroller LPC1768. This microchip provides real time control between a laptop which makes the GUI more advanced. Both of the GUIs allow the user to detect SSID, RSSI, MAC Address and Link Quality of WiFi signals surrounding the laptop. The basic GUI can manually configure the phase shifters and display all of the WiFi information. By changing the phase shifters, it is able to rotate the antenna array main beam direction in order to analyze and monitor the received WiFi signals. Furthermore, this chapter proposes a beta version of an advanced software package for LPC1768, which demonstrates full scanning and basic adaptive beamforming for WiFi signals.

1. The software control system is compatible with Windows XP/Vista/7 and 8, with stable performance.

2. The virtual SPI algorithm is able to accurately configure the digital potentiometers.

3. The GUI for PIC18F4550 is able to generate particular DC voltages and square waveforms for the digital potentiometers to control the smart antenna array.

4. A VB script is developed for PIC18F4550 to achieve an automatic control between the microcontroller and laptop.

5. Real time control is obtained for microprocessor LPC1768 with steady connections.

6. The GUI for LPC1768 demonstrates Manual Control and Automatic Control. Both of the GUI illustrate detailed received WiFi information based on the smart antenna array system

7. The advanced GUI using LPC1768 is able to perform an automatic antenna beam steering and select a best WiFi signal around the control terminal.

3. CONCLUSIONS AND FUTURE WORK

Several preliminary software programs to configure the adaptive beamforming are explored in this paper. In the developed smart antenna, a laptop running Windows is used as a processing device and WiFi signals (2.45GHz) are detected and evaluated. Using a microchip PIC18F4550, the developed graphical user interface is able to manually send various DC voltages and square waveforms to the phase shifters. Moreover, the software programs investigated for the microprocessor LPC1768 can achieve both of manual control and automatic steering. In manual control, it is able to send particular control voltages to the phase shifters and observe the variation of WiFi signals detected by the array. For automatic steering, some basic algorithms are implemented and the software program could perform a full scan first and then select the strongest WiFi signal direction in the environment. This work demonstrates that the fully integrated smart antenna can be applied for future mobile applications.

Overall, it is concluded that the required adaptive antenna performance can be achieved using the fully integrated smart antenna systems. Therefore, it is possible to embed the complete smart antenna into future mobile devices in wireless communication industry.

The future works of this research includes the transfer of GUI into Android and IOS. This research presented several prototype software programs to detect and analyze WiFi signals using the proposed smart antenna in Windows. Future work will continue to improve the control software and also transfer the algorithms into Android and IOS mobile operating systems.

4. REFERENCES

[1] http://www.microchip.com/.
[2] http://orwelldevcpp.blogspot.co.uk/.
[3] http://www.flashmagictool.com/.
[4] G. Wang, M. Romeo, C. Henderson, and J. Papapolymerou, "Novel Reliable RF Capacitive MEMS Switches With Photodefinable Metal-Oxide Dielectrics," Journal of Microelectromechanical Systems, vol. 16, pp. 550-555, 2007.
[5] HMC928LP5E Analog Phase Shifter, Hittite Microwave Corporation.
[6] Y.-G. Kim, K. W. Kim, and Y.-K. Cho, "An ultra-wideband microstrip-to-CPW transition," IEEE MTT-S International Microwave Symposium Digest, pp. 1079-1082, 15-20 June 2008.
[7] D. F. Filipovic, R. F. Bradley, and G. M. Rebeiz, "A planar broadband balanced doubler using a novel balun design," IEEE Microwave and Guided Wave Letters, vol. 4, pp. 229-231, 1994.
[8] PIC18F4550 Data Sheet, Microchip Technology Inc, 2009. [131] LPC1768 Data Sheet, NXP Semiconductors, 2014.
[9] AD5290 Data Sheet, Analog Devices Inc. 2014.
[10] NJM2360 Data Sheet, New Japan Radio Co.Ltd. 2013.
[11] FT232R USB UART I.C. Data Sheet, Future Technology Devices International Ltd. 2005.

BIOGRAPHIES

Aatish Gandotra, a student of Electronics and Communication Engineering in his final year with deep interest in telecommunication and networking.
Publications :- 1. Replication of attacks in wireless sensor network using NS2.

Tejaswi Singh, a student of computer science engineering in his final year with a knack for network security.
Publications :- 1. Replication of attacks in wireless sensor network using NS2.