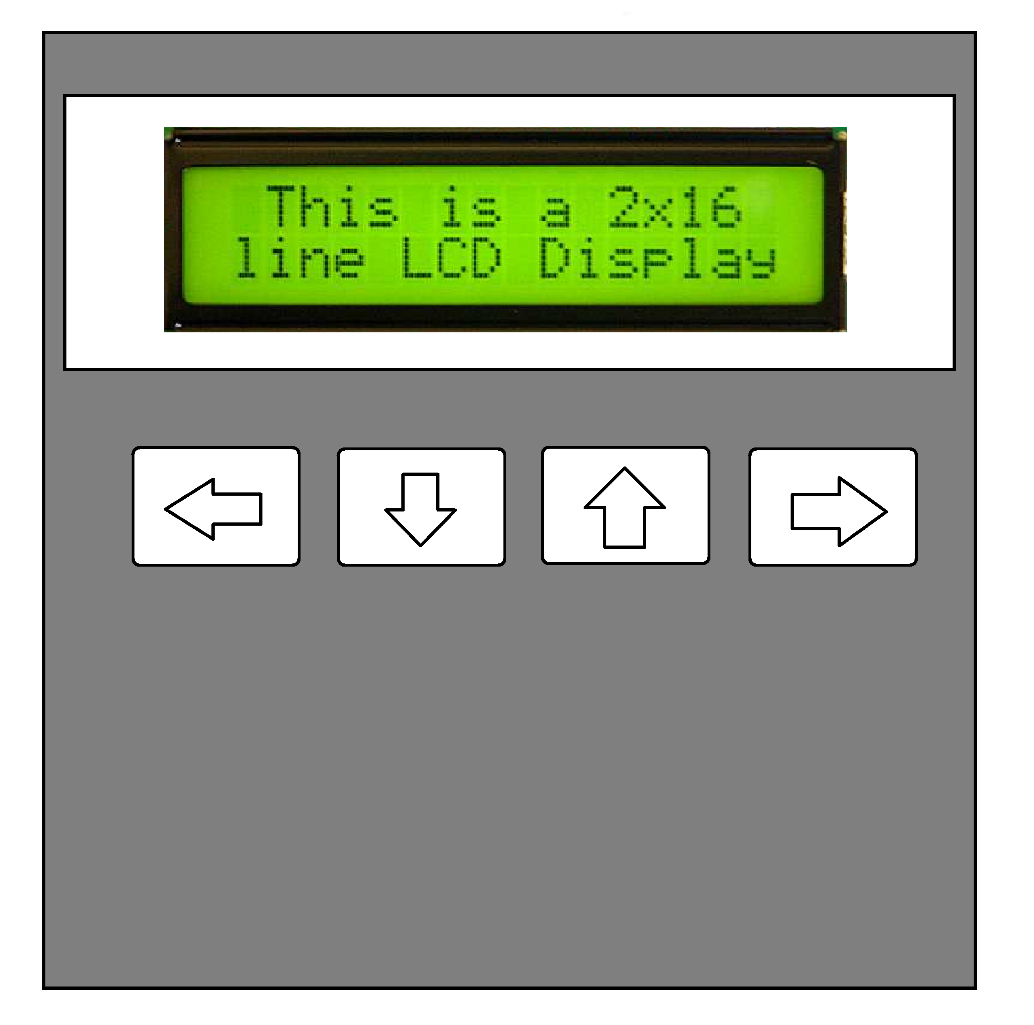
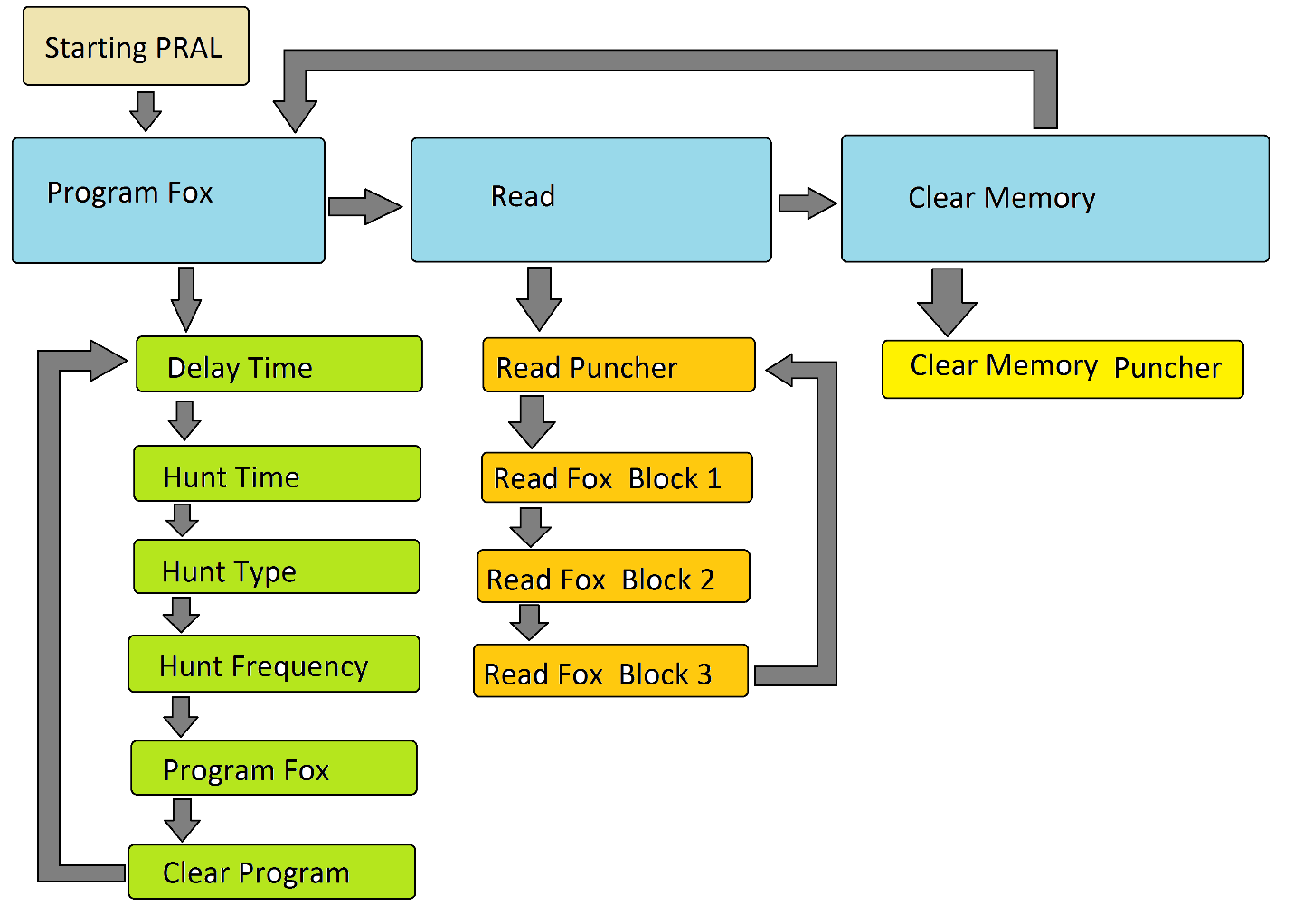
PRAL

The PRAL is a device use to Program the FOX and to read the timing parameters from the FOX and Puncher. Should be used with the Splitter box.



*Fig.1 PRAL Hardware*

After starting the PRAL, the Main Menu is **1.** **Program Fox**, **2.** **Read**, **3.** **Clear Memory**. To jump from one menu to another, the key used are **Left (<-)** and **Right (->)** . For selecting one Option, Press **Long** Right Key **(->) .** In each option there are other options, to jump from one to another use **Left** and **Right** Key **(<- / ->),** for changing the data use **UP** and **DOWN** key. In case of Read, Clear or other actions than changing numbers, use **UP** key. For going back to the Main menu, press **Long** the **Back** key **(<-)** .



*Fig. 2. PRAL Menu organization*

**Program Fox**

The Program Fox menu is used for Programming the Fox.

**Delay time,** used to set the delay time before start of the hunt. The default value is 2 hours. Step time is 10 min. Max time is 42 hours and 30 min.

**Hunt time**, used to set the Hunt period. The default value is 1 hours and 30 min. Step time is 10 min. Max time is 42 hours and 30 min.

**Hunt Type**, for selecting one of the 15 types of the Hunt.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Cycle** | **Foxes** | **Count+Tone** | **ID (sm6rxz)** | **power** | **wpm** | **comment** | **Use** |
| 1 | 1 | 1 | y | y | High | 5 |  | GRJ 1.1 |
| 2 | 2 | 2 | y | y | High | 5 |  | GRJ 1.2 |
| 3 | 3 | 3 | y | y | High | 5 |  | GRJ 1.3 |
| 4 | 4 | 4 | y | y | High | 5 |  | GRJ 1.4 |
| 5 | 5 | 5 | y | y | High | 5 |  | GRJ 1 |
| 6 | 7 | 7 | y | y | High | 5 |  | GRJ 2 |
| 7 | 10 | 10 | y | y | High | 5 |  | GRJ 3 |
| 8 | 5 | 5 | n | y | High | 5 |  | SM Night |
| 9 | 7 | 7 | n | y | High | 5 |  | SM Day |
| 10 | 10 | 10 | n | y | High | 5 |  |  |
| 11 | N/A | X | n | n | Low | 8 | Low power Rules for Foxoring specifies use of 5 to 10 foxes | Beginners training/show, also Foxoring |
| 12 | N/A | X | n | n | High | 5 | High power Rules for Foxoring specifies use of 5 to 10 foxes |  |
| 13 | 1 | 5 | n | n | Low | See comment | 10WPM - 1,2,3,4,5 - Crystal 1 14WPM - 1,2,3,4,5 - Crystal 2 the Foxes 6,7,8,9,10 will be changed (6-1, 7-2, 8-3, 9-4, 10-5)  (each fox transmits 12 second each, in a 1 minute cycle) | Sprint |
| 14 | 2 | 10 | n | n | High | 16 | faster (16wpm), no ID (sm6rxz)  (each fox transmits 12 seconds each, in a 2 minute cycle) | Renberg-hunt (GRJ Sprint) |
| 15 | 5 | 5 | n | n | High | 8 | 6 and 7 send as 4 and 5 - on Crystal2 | National |

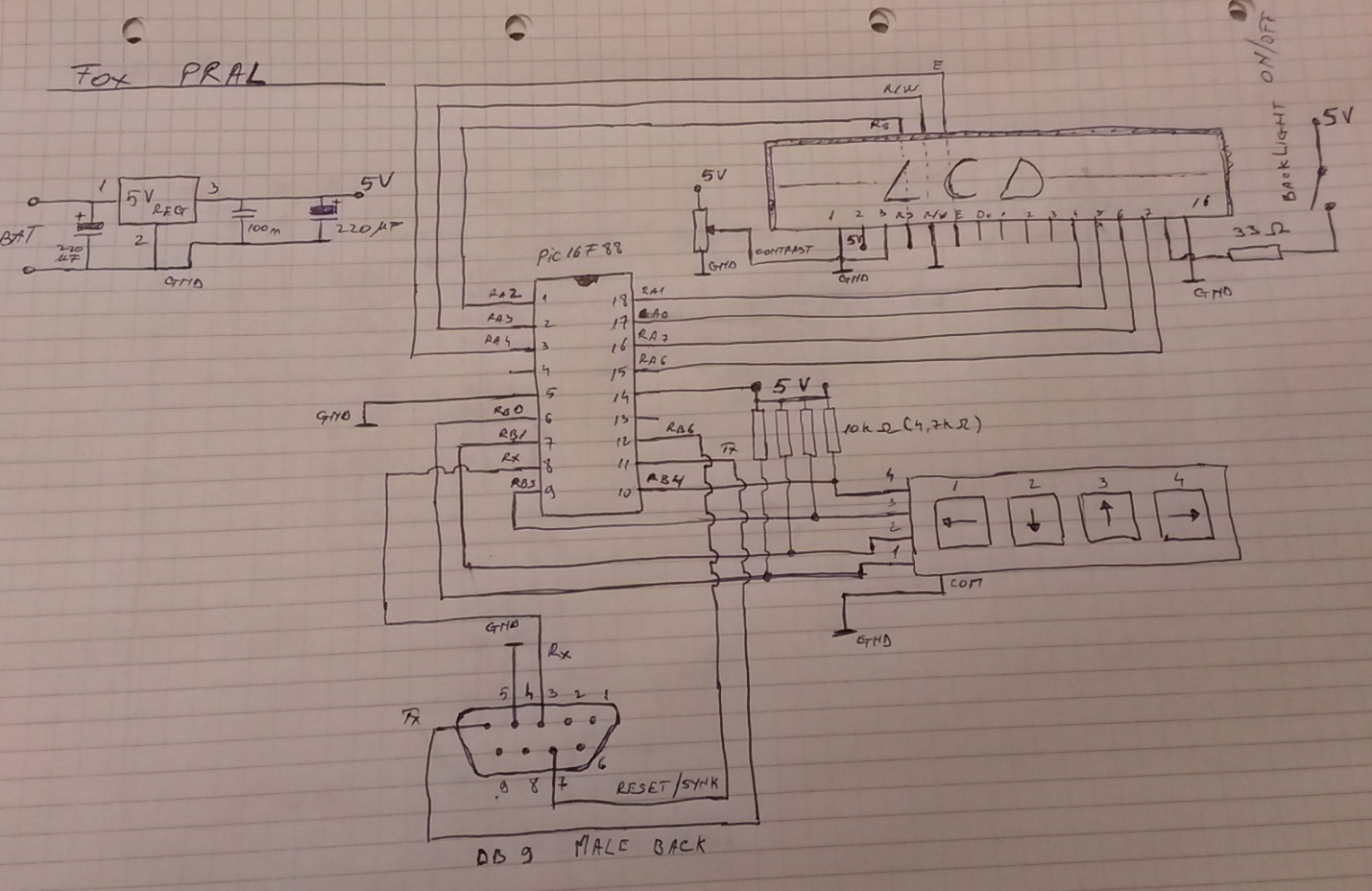
*Tab 1. Hunt Types*

**Hunt frequency,** for selecting the Frequency of the Hunt, can be 3.5 MHz, 144 MHz and Both.

**Progam Fox**, this is the command for Start Programming the Fox. Use the Key **UP** for start sending the data. If the Hunt Organizer want to have the Start time synchronize, than is exactly when is pressing the UP Button. The Programming time is 3 seconds, but if the Delay until start is bigger than 0 then the delay until start will be with 3 sec less. If the Delay until start is 0, the hunt will start 3 sec later, in this case, the **UP** Button must be pressed 3 sec before the desired Starting time.

**Clear Program**, is used Before Program the Fox again, in case of mistake in the previous Programming action. It will reset the Microcontroller. This function is the same as Power Off – On cycle.

**Schematic**

****

**SW Code**

char Select = 0;

char Menu = 0;

char Menu\_Back = 0;

unsigned short Data = 0;

char BTN\_Press\_Flg = 0;

char Block = 0;

unsigned short DelayTime = 12; // 120 min until start

unsigned short Hr\_DelayTime = 0;

unsigned short Mn\_DelayTime = 0;

unsigned short HuntTime = 9; // 90 min Hunt time

unsigned short Hr\_HuntTime = 0;

unsigned short Mn\_HuntTime = 0;

unsigned short HuntType = 1; // Hunt type 1

unsigned short Frequency = 0; // Frequency 3.5

unsigned short i = 0;

unsigned short Prog\_Ent = 0;

char txt[4];

char Eq = '=';

char TP = ':';

char Sel = '-';

char text\_1 = 0; //"Starting"; 8

char text\_2 = 8; //"PRAL"; 4

char text\_3 = 12; //"Delay"; 5

char text\_3\_1 = 17; //"Time"; 4

char text\_4 = 21; //"Hunt"; 4

char text\_3\_4 = 25; //"[x10min]"; 8

char text\_5 = 33; //"Type"; 4

char text\_5\_1 = 37; //"[1-15]"; 6

char text\_6 = 43; //"Frequency"; 9

char text\_7 = 52; //"Program"; 7

char text\_8 = 59; //"Sync"; 4

char text\_10 = 64; //"Done"; 4

char text\_11 = 68; //"Read"; 4

char text\_11\_1 = 72;//"Fox"; 3

char text\_11\_2 = 75;//"Puncher"; 7

char text\_13 = 90; //" 3,5 MHz"; 8

char text\_14 = 98; //" 144 MHz"; 8

char text\_15 = 106; //" Both "; 8

char text\_16 = 114; //"Error"; 5

char text\_17 = 119; //"Clear Memory"; 12

char text\_18 = 144; //"Clear Program"; 13

char text\_19 = 160; //"[H:M]"; 5

char text\_20 = 176; //"Block"; 5

char UART\_Read\_Dta[80]; // = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

char First\_Number\_Read = 0;

char UART\_Read\_Cnt = 0;

char UART\_Rear\_Flg = 0;

unsigned short EEP\_Data;

char Fox\_Nr = 0;

char Hr = 0;

char Min\_ = 0;

char Sec = 0;

// ----------------------------- Inttrerupt ----------------------------------

void Interrupt()

{

UART\_Read\_Dta[UART\_Read\_Cnt]= Usart\_Read();

UART\_Read\_Cnt++;

if(UART\_Read\_Cnt > 79)

{

if(Block == 3) UART\_Read\_Cnt = 0, Block--;

else if(Block == 2) UART\_Read\_Cnt = 0, Block--;

else UART\_Read\_Cnt = 79;

}

if((UART\_Read\_Cnt == 1)&(First\_Number\_Read == 0)) First\_Number\_Read = UART\_Read\_Dta[0], UART\_Read\_Cnt=0;

}

// ---------------------- Display From EEPROM --------------------------------

void Lcd\_Out\_EEP(int Address,int Length,int Row, int Start)

{

for(i=0;i<Length;i++)

{

EEP\_Data = Eeprom\_Read(Address+i);

Lcd\_Chr(Row, (Start+i), EEP\_Data);

}

}

// ---------------------- Display Text --------------------------------

void Lcd\_Out\_Text(int Row\_, int Char\_, int Text\_)

{

ByteToStr(Text\_, txt);

Lcd\_Out(Row\_, Char\_, txt);

}

// ------------------------ Read Buttons ---------------------------------------

void Read\_Button()

{

if ((PORTB.F4 == 0) || (PORTB.F0 == 0))

{

Menu\_Back = Menu;

if(PORTB.F4 == 0) // ->

{

Select++;

delay\_ms(400);

if((PORTB.F4 == 0)& (Menu == 0))

{

delay\_ms(1000);

if(PORTB.F4 == 0) Menu = Select, Select=0;

}

}

else if(PORTB.F0 == 0) // <-

{

Select--;

delay\_ms(400);

if((PORTB.F0 == 0)& (Menu > 0))

{

delay\_ms(1000);

if(PORTB.F0 == 0) Select=(Menu-1), Menu = 0 ;

}

}

Data=0;

UART\_Rear\_Flg = 0;

Prog\_Ent = 0;

Lcd\_Cmd(LCD\_CLEAR);

// ------------------------- Function Menu Display ------------------

// --------------------- 1. Program -----------

if(Menu == 1)

{

if(Select > 200) Select=5;

else if(Select > 5) Select=0;

if(Select == 0) // Delay Time

{

Lcd\_Chr(1, 1, '1');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_3,5,1,4);

Lcd\_Out\_EEP(text\_3\_1,4,1,10);

Lcd\_Out\_EEP(text\_19,5,2,12);

Lcd\_Chr(2, 1, Eq);

if(DelayTime > 0)Data = DelayTime;

}

else if(Select == 1) // Hunt Time

{

Lcd\_Chr(1, 1, '2');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_4,4,1,4);

Lcd\_Out\_EEP(text\_3\_1,4,1,9);

Lcd\_Out\_EEP(text\_19,5,2,12);

Lcd\_Chr(2, 1, Eq);

if(HuntTime > 0) Data = HuntTime;

}

else if(Select == 2) // Hunt Type

{

Lcd\_Chr(1, 1, '3');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_4,4,1,4);

Lcd\_Out\_EEP(text\_5,4,1,9);

Lcd\_Out\_EEP(text\_5\_1,6,2,11);

Lcd\_Chr(2, 1, Eq);

if(HuntType > 0) Data = HuntType;

}

else if(Select == 3) // Hunt Frequency

{

Lcd\_Chr(1, 1, '4');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_6,9,1,4);

Lcd\_Chr(2, 1, Eq);

if(Frequency > 0) Data = Frequency;

}

else if(Select == 4) // Program Fox

{

Lcd\_Chr(1, 1, '5');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_7,7,1,4);

Lcd\_Out\_EEP(text\_11\_1,3,1,12);

}

else if(Select == 5) // Clear Program

{

Lcd\_Chr(1, 1, '6');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_18,13,1,4);

}

}

// ------------------ 2. Read -----------

else if(Menu == 2)

{

if(Select > 200) Select=3;

if(Select > 3) Select=0;

if(Select == 0) // Read Puncher

{

Lcd\_Chr(1, 1, '1');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_11,4,1,4);

Lcd\_Out\_EEP(text\_11\_2,7,1,9);

}

else if(Select == 1) // Read Fox Block 1

{

Lcd\_Chr(1, 1, '2');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_11,4,1,4);

Lcd\_Out\_EEP(text\_11\_1,3,1,9);

Lcd\_Out\_EEP(text\_20,2,1,13);

Lcd\_Chr(1, 16, '1');

Block = 1;

}

else if(Select == 2) // Read Fox Block 2

{

Lcd\_Chr(1, 1, '3');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_11,4,1,4);

Lcd\_Out\_EEP(text\_11\_1,3,1,9);

Lcd\_Out\_EEP(text\_20,2,1,13);

Lcd\_Chr(1, 16, '2');

Block = 2;

}

else if(Select == 3) // Read Fox Block 3

{

Lcd\_Chr(1, 1, '4');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_11,4,1,4);

Lcd\_Out\_EEP(text\_11\_1,3,1,9);

Lcd\_Out\_EEP(text\_20,2,1,13);

Lcd\_Chr(1, 16, '3');

Block = 3;

}

}

// ------------------- 3. Clear Memory ----

else if(Menu == 3) //Clear Memory

{ //Clear Puncher

Lcd\_Chr(1, 1, '1');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_17,5,1,4);

Lcd\_Out\_EEP(text\_11\_2,7,1,10);

}

}

// ------------------------- Main Menu Display ---------------------------------

if(Menu == 0)

{

if(Select >200) Select=2;

else if (Select >2) Select=0;

if(Select == 0)

{

Lcd\_Chr(1, 1, '1');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_7,7,1,4); // Program Fox

}

else if(Select == 1)

{

Lcd\_Chr(1, 1, '2');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_11,4,1,4); //Read

}

else if(Select == 2)

{

Lcd\_Chr(1, 1, '3');

Lcd\_Chr(1, 2, '.');

Lcd\_Out\_EEP(text\_17,12,1,4); //"Clear Memory"; 12 Fox

}

}

// ---------- Read buttons UP - Down -----------------------------

if(BTN\_Press\_Flg == 1) delay\_ms(500), BTN\_Press\_Flg=0;

if(PORTB.F3 == 0) // Up

{

Data++;

if((Menu == 1) & (Select ==4)&(Prog\_Ent==0)) Prog\_Ent = 1;

BTN\_Press\_Flg = 1;

}

else if(PORTB.F1 == 0) // Down

{

Data--;

BTN\_Press\_Flg = 1;

}

}

// ==================== Start Main Program =====================================

void main() {

TRISA = 0x00;

TRISB = 0x1F;

ANSEL = 0x00;

ADCON0 = 0x00;

OSCCON =0x7A;

Lcd\_Config(&PORTA, 2, 4, 3, 6, 7, 0, 1); // Initialize LCD on PORTB

Lcd\_Cmd(LCD\_CURSOR\_OFF);

Usart\_Init(4800);

INTCON = 0xC0; // Activate Interrupt - PEIE: Peripheral Interrupt Enable bit

PIE1 = 0x20; // RCIE: EUSART Receive Interrupt Enable

Lcd\_Out\_EEP(text\_1,8,1,5);

delay\_ms(500);

Lcd\_Chr(2, 6, Sel);

Lcd\_Out\_EEP(text\_2,4,2,7);

Lcd\_Chr(2, 11, Sel);

delay\_ms(1500);

Lcd\_Cmd(LCD\_CLEAR);

while(1)

{

Read\_Button();

// ------------------------------ Functions -----------------------------

// ----------------------------- Program -------------------------------

if(Menu == 1)

{

if(Select == 0) // Delay Time

{

DelayTime = Data;

Hr\_DelayTime = (DelayTime/6);

Mn\_DelayTime = ((DelayTime%6)\*10);

Lcd\_Out\_Text(2,2,Hr\_DelayTime);

Lcd\_Chr(2,5,TP);

Lcd\_Out\_Text(2,6,Mn\_DelayTime);

}

else if(Select == 1) // Hunt Time

{

HuntTime = Data;

Hr\_HuntTime = (HuntTime/6);

Mn\_HuntTime = ((HuntTime%6)\*10);

Lcd\_Out\_Text(2,2,Hr\_HuntTime);

Lcd\_Chr(2,5,TP);

Lcd\_Out\_Text(2,6,Mn\_HuntTime);

}

else if(Select == 2) // Hunt Type

{

if(Data > 15) Data = 1;

else if(Data < 1) Data = 15;

HuntType = Data;

Lcd\_Out\_Text(2,2,HuntType);

}

else if(Select == 3) // Hunt Frequency

{

if(Frequency > 2) Data=0;

Frequency = Data;

if(Frequency == 0) Lcd\_Out\_EEP(text\_13,8,2,3);

else if(Frequency == 1) Lcd\_Out\_EEP(text\_14,8,2,3);

else if(Frequency == 2) Lcd\_Out\_EEP(text\_15,8,2,3);

}

else if(Select == 4) // Program Fox

{

if(Prog\_Ent == 1)

{

UART\_Read\_Cnt = 0;

Lcd\_Chr(2, 1, 0xFF); delay\_ms(70);

Usart\_Write(0x50);

Lcd\_Chr(2, 2, 0xFF); delay\_ms(70);

Usart\_Write(0x3F);

Lcd\_Chr(2, 3, 0xFF); delay\_ms(70);

Usart\_Write(DelayTime);

Lcd\_Chr(2, 4, 0xFF); delay\_ms(70);

Usart\_Write(HuntTime);

Lcd\_Chr(2, 5, 0xFF); delay\_ms(70);

Usart\_Write(HuntType);

Lcd\_Chr(2, 6, 0xFF); delay\_ms(70);

Usart\_Write(Frequency);

Lcd\_Chr(2, 7, 0xFF); delay\_ms(80);

Usart\_Write(0x00); // Clear Hunt Flag

delay\_ms(2500);

Lcd\_Out\_EEP(text\_10,4,2,9);

Prog\_Ent = 2;

}

}

else if(Select == 5) // Clear Program

{

if(Data > 0)

{

for(i=1;i<9;i++) Lcd\_Chr(2, i, ' ');

UART\_Read\_Cnt = 0;

Usart\_Write(0x50);

delay\_ms(30);

Usart\_Write(0x4F);

Data=0;

delay\_ms(50);

Lcd\_Chr(2, 1, 0xFF);

Lcd\_Out\_EEP(text\_10,4,2,2);

}

}

}

// -------------------------- Read --------------------------------------------

if(Menu == 2)

{

// ------------------- Read Puncher -----------------

if(Select == 0)

{

if((Data > 0) & (UART\_Rear\_Flg == 0))

{

Block = 0;

First\_Number\_Read=0;

UART\_Read\_Cnt = 0;

Usart\_Write(0x60);

Usart\_Write(0x2F);

Usart\_Write(0x00);

Usart\_Write(0x01);

for(i=1;i<12;i++)

{

Lcd\_Chr(2, i, 0xFF);

delay\_ms(190);

}

Lcd\_Out\_EEP(text\_10,4,2,13);

delay\_ms(200);

UART\_Rear\_Flg = 1;

Lcd\_Cmd(LCD\_CLEAR);

Data=20;

}

else if (UART\_Rear\_Flg == 1)

{

Lcd\_Out\_EEP(text\_11\_2,7,1,1);

Lcd\_Chr(1,8,TP);

Lcd\_Out\_Text(1,10,First\_Number\_Read);

if(Data > 200) Data = 0;

else if(Data > 20) Data = 20;

Fox\_Nr = UART\_Read\_Dta[(4\*(20-Data))];

Hr = UART\_Read\_Dta[(4\*(20-Data))+1];

Min\_ = UART\_Read\_Dta[(4\*(20-Data))+2];

Sec = UART\_Read\_Dta[(4\*(20-Data))+3];

Lcd\_Out\_Text(2,1,Fox\_Nr);

Lcd\_Chr(2,4,'-');

Lcd\_Out\_Text(2,5,Hr);

Lcd\_Chr(2,8,TP);

Lcd\_Out\_Text(2,9,Min\_);

Lcd\_Chr(2,12,TP);

Lcd\_Out\_Text(2,13,Sec);

}

}

// -------------------------- Read Fox -----------------------------

else

{

if((Data > 0) & (UART\_Rear\_Flg == 0))

{

First\_Number\_Read=0;

UART\_Read\_Cnt = 0;

Usart\_Write(0x50);

Usart\_Write(0x2F);

for(i=1;i<12;i++)

{

Lcd\_Chr(2, i, 0xFF);

delay\_ms(190);

}

Lcd\_Out\_EEP(text\_10,4,2,13);

delay\_ms(200);

UART\_Rear\_Flg = 1;

Lcd\_Cmd(LCD\_CLEAR);

Data=20;

}

else if (UART\_Rear\_Flg == 1)

{

Lcd\_Out\_EEP(text\_11\_1,3,1,1);

Lcd\_Chr(1,4,TP);

Lcd\_Out\_Text(1,6,First\_Number\_Read);

if(Data > 200) Data = 0;

else if(Data > 20) Data = 20;

Fox\_Nr = UART\_Read\_Dta[(4\*(20-Data))];

Hr = UART\_Read\_Dta[(4\*(20-Data))+1];

Min\_ = UART\_Read\_Dta[(4\*(20-Data))+2];

Sec = UART\_Read\_Dta[(4\*(20-Data))+3];

Lcd\_Out\_Text(2,1,Fox\_Nr);

Lcd\_Chr(2,4,'-');

Lcd\_Out\_Text(2,5,Hr);

Lcd\_Chr(2,8,TP);

Lcd\_Out\_Text(2,9,Min\_);

Lcd\_Chr(2,12,TP);

Lcd\_Out\_Text(2,13,Sec);

}

}

}

//------------------------Clear Memory Puncher ---------------------------------

if(Menu == 3)

{

if(Data > 0)

{

for(i=1;i<17;i++) Lcd\_Chr(2, i, ' ');

First\_Number\_Read=0;

UART\_Read\_Cnt = 0;

UART\_Read\_Dta[0] = 0x00;

Usart\_Write(0x60);

delay\_ms(50);

Usart\_Write(0xFF);

delay\_ms(50);

Usart\_Write(0x00);

delay\_ms(50);

Usart\_Write(0xFF);

Data=0;

for(i=1;i<12;i++)

{

Lcd\_Chr(2, i, 0xFF);

delay\_ms(200);

}

if(First\_Number\_Read == 0x6F) Lcd\_Out\_EEP(text\_10,4,2,13);

else Lcd\_Out\_EEP(text\_16,5,2,12);

}

}

// ----------- Wait after enter in each Menu -----------------------------------

if(Menu\_Back != Menu)

{

delay\_ms(700);

Menu\_Back = Menu;

}

} // End while

}