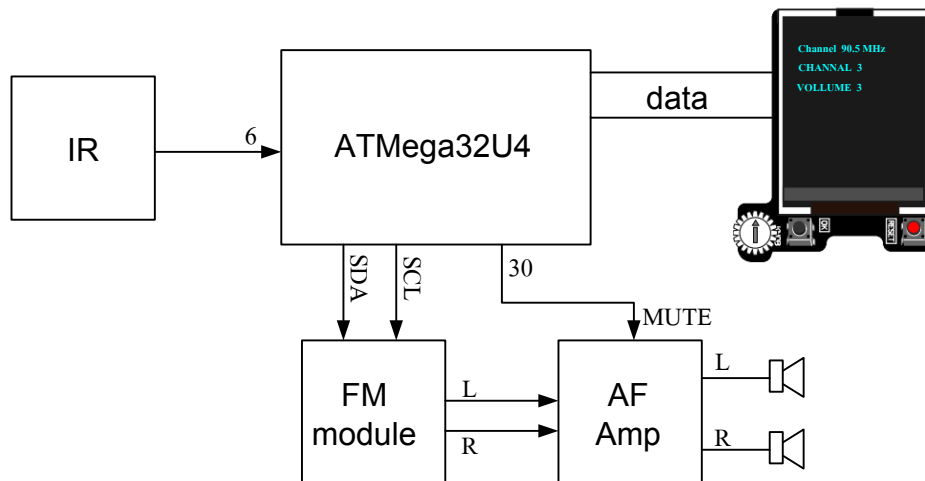


41. นำไมโครคอนโทรลเลอร์ ความคุมโมดูลเครื่องรับวิทยุ และควบคุมเครื่องขยาย



```
#include <unicon.h>

#define SLAVE_ID 0x20 >> 1

#define CONFIG_WORDS 5

#include <IRremote.h>

int RECV_PIN = 6;

int AMP = 30;

int N;

IRrecv irrecv(RECV_PIN);

decode_results results;

static uint16_t Config0[CONFIG_WORDS] = {0xD001, 0x0000, 0x0400, 0x86D3, 0x4000};

static uint16_t Config[CONFIG_WORDS];

#define SEEK 8

#define SEEK_MASK (3 << SEEK)

#define SEEK_UP (3 << SEEK)

#define SEEK_DOWN (1 << SEEK)

#define CHAN 6

#define TUNE 4

#define CHAN_MASK (0xff << CHAN)

#define VOLUME 0
```

```
#define VOLUME_MASK (0x0f << VOLUME)

#define STC 14

#define READCHAN 0

#define READCHAN_MASK (0xff << READCHAN)

#define MAX_FREQ 1080

#define MIN_FREQ 870

int vol_max=15;

int t1=0;

uint16_t fm_vol = 1, fm_chan = 965;

uint16_t fm_stc, fm_seek;

bool seeking;

int pinFreqDown = 18;

int pinFreqUp = 19;

int pinSeekDown = 20;

int pinSeekUp = 21;

void setup()

{

    pinMode(pinFreqDown, INPUT);

    pinMode(pinFreqUp, INPUT);

    pinMode(pinSeekDown, INPUT);

    pinMode(pinSeekUp, INPUT);

    pinMode(AMP, OUTPUT);

    glcdClear();

    irrecv.enableIRIn();

    Wire.begin();

    fm_reset();

    fm_reset();

    chan_print();
```

```

fm_vol == 1;

glcd(4, 1, "CHANNAL 4  ");

glcd(6, 0, " VOLUME  1");
}

void loop()
{
  if (irrecv.decode(&results))
  {

    if(results.value == 1168 && fm_vol < 10)
    {
      fm_vol = fm_vol + 1;

      fm_reset();

      glcd(6, 0, " VOLUME  ", "%d  ", fm_vol);

      glcd(6, 11, "%d  ", fm_vol);
    }

    else if (results.value == 3216 && fm_vol > 0)
    {
      fm_vol = fm_vol - 1;

      fm_reset();

      glcd(6, 0, " VOLUME  ", "%d  ", fm_vol);

      glcd(6, 11, "%d  ", fm_vol);
    }

    else if (results.value == 656)
    {
      if(t1) t1=0;

```

```

else t1 =1;

if(t1)
{
digitalWrite(AMP, HIGH);

glcd(6, 0, " VOLUME  0  ");

}

else
{
digitalWrite(AMP, LOW);

glcd(6, 0, " VOLUME  ", "%d  ", fm_vol);

glcd(6, 11, "%d  ", fm_vol);

}

}

else if (results.value == 2704)
{
if(t1) t1=0;

else t1 =1;

if(t1)

ONOFF ();

else

OFFON ();

}

if(results.value == 16)
{

CH1 ();

}

else if(results.value == 2064)

```

```
    {  
        CH2 ();  
    }  
else if(results.value == 1040)  
    {  
        CH3 ();  
    }  
else if(results.value == 3088)  
    {  
        CH4 ();  
    }  
else if(results.value == 528)  
    {  
        CH5 ();  
    }  
else if(results.value == 2576)  
    {  
        CH6 ();  
    }  
else if(results.value == 1552)  
    {  
        CH7 ();  
    }  
else if(results.value == 3600)  
    {  
        CH8 ();  
    }  
else if(results.value == 272)
```

```
{
    CH9 ();
}
else if(results.value == 2320)
{
    CH10 ();
}
else if(results.value == 3280)
{
    SEEKUP ();
}
else if(results.value == 720)
{
    SEEKDOWN ();
}
else if(results.value == 752)
{
    FREQUP ();
}
else if(results.value == 2800)
{
    FREQDOWN ();
}
if(results.value == 144 && N < 10)
{
    N = N + 1;
    CHKCH ();
}
```

```
    if(results.value == 2192 && N > 1)
    {
        N = N - 1;
        CHKCH ();
    }
    irrecv.resume();
}

if(!digitalRead(pinFreqDown))
{
    FREQDOWN ();
}

else if(!digitalRead(pinFreqUp))
{
    FREQUP ();
}

else if(!digitalRead(pinSeekDown))
{
    SEEKDOWN ();
}

else if(!digitalRead(pinSeekUp))
{
    SEEKUP ();
}

if (seeking)
{
    setConfig();
    delay(100);
}
```

```

    getStatus();
    if (seeking && fm_stc)
    {
        seek_disable();
        chan_update(fm_seek + MIN_FREQ);
        setConfig();
    }
    tune_disable();
}

```

```

void setConfig()

```

```

{
    Wire.beginTransaction(SLAVE_ID);
    for(int i = 0 ; i < CONFIG_WORDS ; i++)
    {
        Wire.write((int)Config[i] >> 8);
        Wire.write((int)(Config[i] & 0xff));
    }
    Wire.endTransmission();
}

```

```

void fm_reset()

```

```

{
    seeking = false;
    memcpy(Config,Config0,CONFIG_WORDS * sizeof(uint16_t));
    Config[1] &= ~CHAN_MASK;
    Config[1] |= (fm_chan - MIN_FREQ) << CHAN;
}

```



```
Config[3] &= ~VOLUME_MASK;

Config[3] |= fm_vol;

tune_enable();

setConfig();

tune_disable();

}
```

```
void tune_enable()

{

    Config[1] |= (1 << TUNE);

}
```

```
void tune_disable()

{

    Config[1] &= ~(1 << TUNE);

}
```

```
void seek_disable()

{

    seeking = false;

    Config[0] &= ~SEEK_MASK;

}
```

```
void chan_update(int val)

{

    if (val >= MIN_FREQ || val <= MAX_FREQ)

    {

        fm_chan = val;

    }

}
```

```

    Config[1] &= ~CHAN_MASK;

    Config[1] |= (fm_chan - MIN_FREQ) << CHAN;

    tune_enable();

    chan_print();

}

}

```

```
void chan_print()
```

```

{

    int chan = ((Config[1] & CHAN_MASK) >> CHAN) + MIN_FREQ;

    glcd(2, 0, " Channel %d.%d MHz ", chan / 10, chan % 10);

}

```

```
void getStatus()
```

```

{

    uint16_t Status[1];

    Wire.requestFrom(SLAVE_ID, 2);

    for(int i = 0 ; i < 1 ; i++)

        {

            Status[i] = Wire.read();

            Status[i] = Status[i] << 8 | Wire.read();

        }

    fm_seek = Status[0] & READCHAN_MASK;

    fm_stc = Status[0] >> STC & 1;

}

```

```
void ONOFF()
```

```
{  
    digitalWrite(AMP, HIGH);  
    glcdClear ();  
}
```

```
void OFFON()
```

```
{  
    CHKCH ();  
    fm_vol = 1;  
    fm_reset();  
    chan_print();  
    digitalWrite(AMP, LOW);  
    glcd(6, 0, " VOLUME 1");  
}
```

```
void CH1 ()
```

```
{  
    N = 1;  
    fm_chan = 880;  
    fm_reset();  
    chan_print();  
    glcd(0, 0, " ");  
    glcd(4, 1, "CHANNAL 1 ");  
}
```

```
void CH2 ()
```

```
{  
    N = 2;  
    fm_chan = 900;  
    fm_reset();
```

```

        chan_print();

        glcd(0, 0, "          ");

        glcd(4, 1, "CHANNAL 2  ");
    }

void CH3 ()
{
    N = 3;

    fm_chan = 905;

    fm_reset();

    chan_print();

    glcd(0, 0, "          ");

    glcd(4, 1, "CHANNAL 3  ");
}

void CH4 ()
{
    N = 4;

    fm_chan = 965;

    fm_reset();

    chan_print();

    glcd(0, 0, "          ");

    glcd(4, 1, "CHANNAL 4  ");
}

void CH5 ()
{
    N = 5;

    fm_chan = 970;

    fm_reset();

    chan_print();

```

```

        glcd(0, 0, "        ");
        glcd(4, 1, "CHANNAL 5    ");
    }
void CH6 ()
{
    N = 6;
    fm_chan = 980;
    fm_reset();
    chan_print();
    glcd(0, 0, "        ");
    glcd(4, 1, "CHANNAL 6    ");
}
void CH7 ()
{
    N = 7;
    fm_chan = 1040;
    fm_reset();
    chan_print();
    glcd(0, 0, "        ");
    glcd(4, 1, "CHANNAL 7    ");
}
void CH8 ()
{
    N = 8;
    fm_chan = 1042;
    fm_reset();
    chan_print();
    glcd(0, 0, "        ");
}

```

```

        glcd(4, 1, "CHANNAL 8  ");
    }
void CH9 ()
{
    N = 9;
    fm_chan = 1050;
    fm_reset();
    chan_print();
    glcd(0, 0, "          ");
    glcd(4, 1, "CHANNAL 9  ");
}
void CH10 ()
{
    N = 10;
    fm_chan = 1060;
    fm_reset();
    chan_print();
    glcd(0, 0, "          ");
    glcd(4, 1, "CHANNAL 10  ");
}
void FREQUPO
{
    glcd(0, 0, "FREQ UP  ");
    glcd(4, 1, "          ");
    delay(100);
    chan_update(fm_chan + 1);
    seeking = true;
    chan_print();
}

```

```
void FREQDOWN()
```

```
{  
    glcd(0, 0, " FREQ DOWN ");  
    glcd(4, 1, " ");  
    delay(100);  
    chan_update(fm_chan - 1);  
    seeking = true;  
    chan_print();  
}
```

```
void SEEKDOWN()
```

```
{  
    glcd(0, 0, " SEEK DOWN ");  
    glcd(4, 1, " ");  
    delay(100);  
    Config[0] &= ~SEEK_MASK;  
    Config[0] |= SEEK_DOWN;  
  
    seeking = true;  
    chan_print();  
}
```

```
void SEEKUP()
```

```
{  
    glcd(0, 0, " SEEK UP ");  
    glcd(4, 1, " ");  
    delay(100);  
    Config[0] &= ~SEEK_MASK;  
    Config[0] |= SEEK_UP;  
    seeking = true;
```

```
        chan_print();
    }
void CHKCH ()
{
    if(N == 1)
        CH1 ();
    else if (N == 2)
        CH2 ();
    else if (N == 3)
        CH3 ();
    else if (N == 4)
        CH4 ();
    else if (N == 5)
        CH5 ();
    else if (N == 6)
        CH6 ();
    else if (N == 7)
        CH7 ();
    else if (N == 8)
        CH8 ();
    else if (N == 9)
        CH9 ();
    else if (N == 10)
        CH10 ();
}
```