

DMR-5WUF

The module	3
Module size and pin assignment	3
Typical Application Circuit Diagram	4
Serial protocol	5
Packet format	5
Checksum calculate:	7
Commands	8
0x22: Set digital channel	8
0x23: Set analog channel	10
0x24: query digital channel information	11
0x25: query analog channel information	11
0x26: Set transmit information	12
0x27: query if the module finished the initialization	13
0x28: Set enhanced function	14
0x29: Set encryption function (inte tillåtet för amatörbruk)	15
0x2A: Set MICgain	16
0x2B: Query digital voice receive information	17
0x2C: send SMS	18
0x2D: get SMS	19
0x2E: Set Volume	21
0x2F: Set monitor	22
0x30: Set squelch (analogt)	23
0x31: Set power save mode	24
0x32: query signal strenght	25
0x33: Set relay disconnect net mode	26
0x34: Query version	27
0x3C: Pause transmit function	28
0x3D: Get message	29
Voice transmission and receive	30
Voice transmission	30
Voice receive flowchart	32
Short message communication (SMS)	34
UnacknowledgedSMS sending flowchart	34
UnacknowledgedSMS receive flowchart	35
AcknowledgedSMS Sending flowchart	36
AcknowledgedSMS Receive flowchart	37
APPENDIX	38

UART checksum calculate	38
arduino kodexempel	40
Arduino objekts	40
Checksummeexempel	41
Samlade erfarenheter	42
Kanalnummer	46
Termer för kanaler Anytone och DMR module	47
Projektlogg DMR enheten	49
Tabeller som behöver hanteras internt	56
Hotspot konfiguration	57

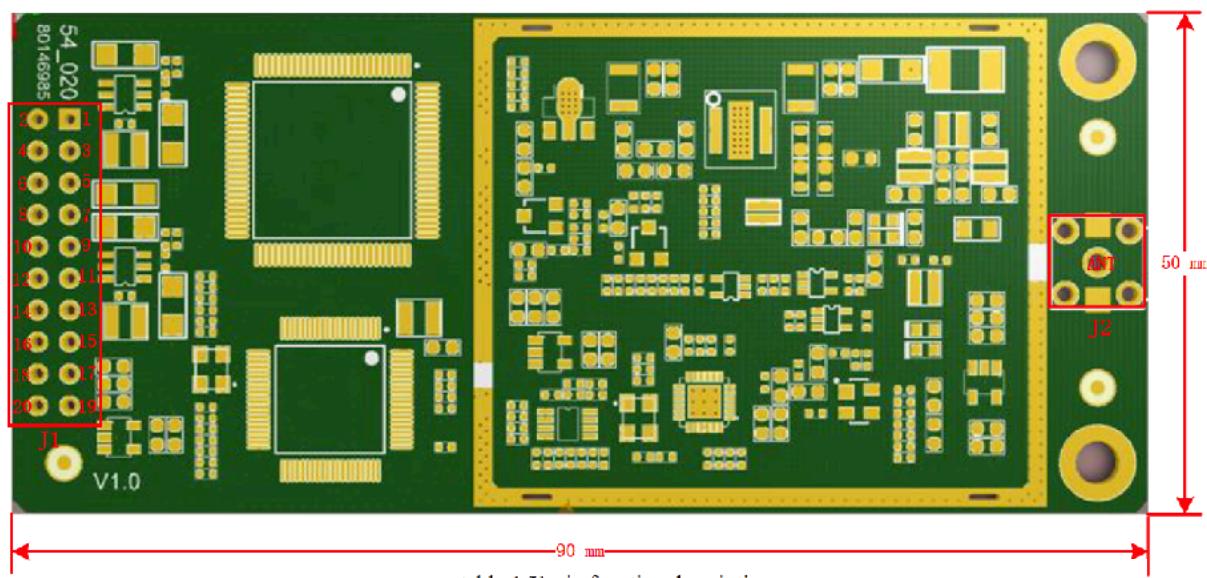
The module

Module size and pin assignment

Size: 50mm. 90mm.

J1: MCU interface

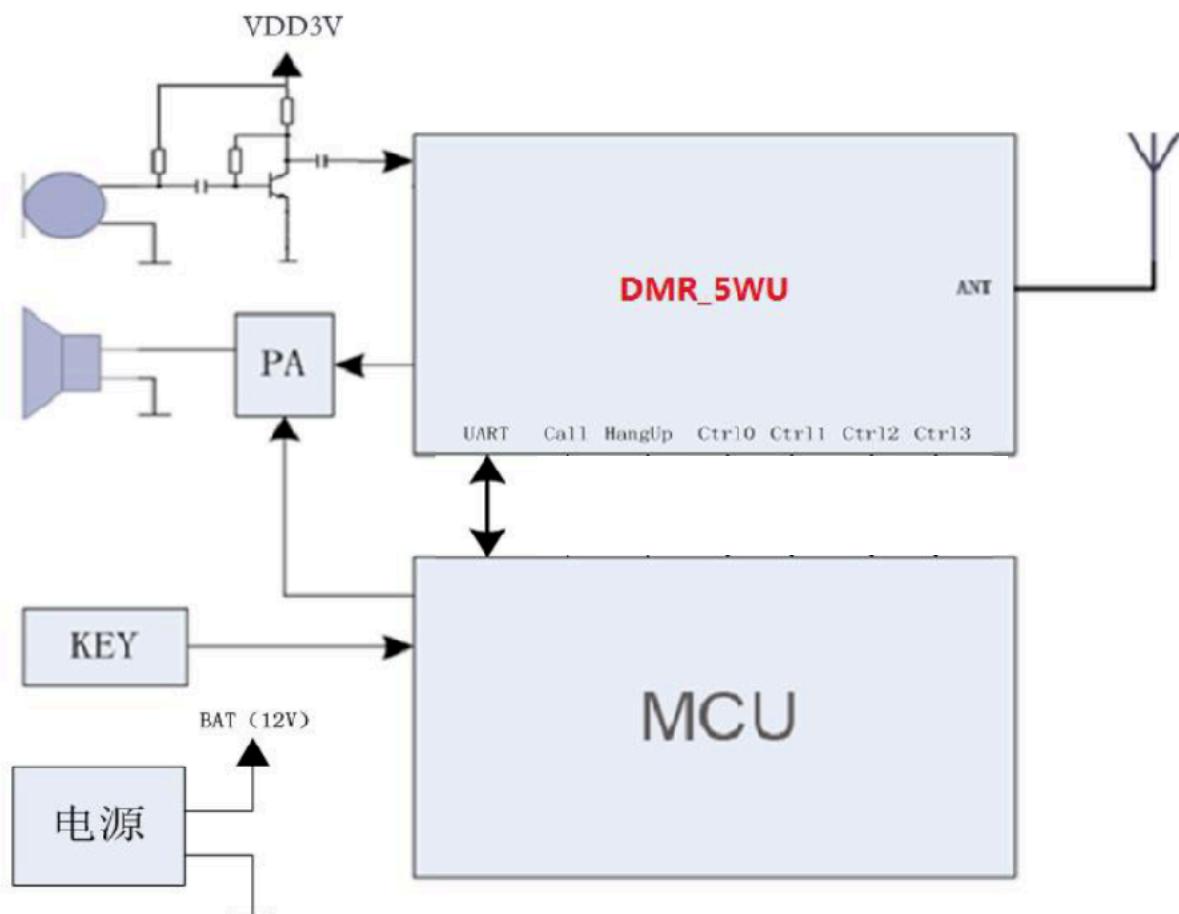
J2: Ant jack



Pin num	Pin name	type	description
1	VBAT	POWER	DC power
2	VBAT	POWER	DC power
3	GND	GND	GND
4	GND	GND	GND
5	UART_TX	DO	TXD for UART
6	UART_RX	DI	RXD for UART
7	HANGUP	DI	Accept Calling ; Low puls > 20ms
8	CALL	DI	Calling, Low puls > 20ms
9	CTRL_D0	DIO	reserved
10	CTRL_D1	DIO	reserved
11	CTRL_D2	DIO	reserved
12	CTRL_D3	DIO	reserved
13	GND	GND	GND
14	NC	NC	NC
15	MIC_IN	AI	Mic input
16	GND	GND	GND
17	LINEOFOUT	AO	Audio output
18	GND	GND	GND
19	GND	GND	GND
20	GND	GND	GND

note1: when power on, CALL, HANGUP pin must be pull high;

Typical Application Circuit Diagram



5 Technical Parameter

5.1 Electrical Characteristics

表 2 电气特性

parameter	condition	MIN	Typical	Max	Unit
Power Supply		9	12	18	V
temperature		-20		60	℃
Startup Time of Module		100			ms
Series Baut Rate			56700		bps
Input voltage of microphone		0.5		1	Vpp
Lineout output				1	Vpp

Serial protocol

Packet format

Modulen stöder sändtagarens konfiguration av röst, SMS och andra funktioner via den seriella porten. Paketformatet för seriell portprotokoll visas i figur 6

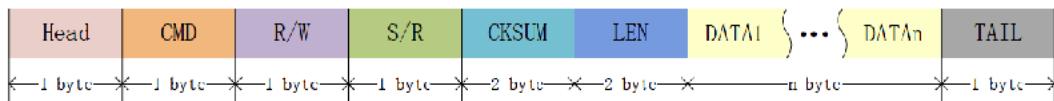


图 6 串口协议包格式

表 4 串口协议字段定义

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x22: Set digital channel 0x23: Set analog channel 0x24: query digital channel information 0x25: query analog channel information 0x26: Set transmit information 0x27: query if the module finished the initialization 0x28: Set enhanced function 0x29: Set encryption function 0x2A: Set MIC gain 0x2B: query digital voice receive

				information 0x2C: send SMS 0x2D: get SMS 0x2E: Set Volume 0x2F: Set monitor 0x30: Set squelch 0x31: Set power save mode 0x32: query signal strength 0x33: Set relay disconnect net mode 0x34: query version 0x3C: pause transmit function
2	R/W	1	operation type	0x00: read : 0x01: write : (external CPU sending is writing, external CPU receive is reading) 0x02: forwardly send

3	S/R	1	Set / response	Set : 0x01: Setting response : 0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	
6, 7	LEN	2	data length	DATA leng, if no data information , then LEN is 0x00
8	DATA	len	data information	
	TAIL	1	end package	0x10

LEN första byten är alltid 0. Dvs längd en byte är t.ex 00 01.

```
{
    uint8_t head;           // header 0x68
    uint8_t CMD;            // command 0x22
    uint8_t RW;              // operation type
    uint8_t SR;              // set/response
    uint16_t CKSUM;          // checksum
    uint8_t LEN_dummy;        // måste vara = 0
    uint8_t LEN;              // DATA length
}
```

Checksum calculate:

```
int16_t PcCheckSum(uint8_t * buf, int len)
{
    uint32_t sum=0;
    while(len >1)
    {
        sum += 0xFFFF & (*buf<<8|*(buf+1));
        buf+=2;
        len-=2;
    }
    if (len)
    {
        sum += (0xFF & *buf)<<8;
    }
    while (sum>>16)
    {
        sum = (sum & 0xFFFF)+(sum >> 16);
    }
    return( (uint16_t) sum ^ 0xFFFF);
}
```

Little Endian mode

<https://searchnetworking.techtarget.com/definition/big-endian-and-little-endian>

Commands

0x22: Set digital channel

```
struct db_digital_info
{
    uint8_t head;           //header 0x68
    uint8_t CMD;            // command 0x22
    uint8_t RW;              // operation type
    uint8_t SR;              // set/response
    uint16_t CKSUM;          // checksum
    uint8_t LEN_dummy;        // måste vara = 0
    uint8_t LEN;              // DATA length
    uint32_t rx_freq;         //receive frequency 400000000-48000000HZ
    uint32_t tx_freq;         //transmit frequency 400000000-48000000HZ
    uint32_t localID;         //local ID1-16776415
    uint32_t GroupList[32];   //receive group list
    uint32_t tx_contact;      //contact number 1-16776415
    uint8_t ContactType;       //contact type 0:individual call 1:group call2:call all 3: full duplex
    uint8_t power;             //0:low power 1: high power
    uint8_t cc;                //color code0~15
    uint8_t InboundSlot;       //0:slot 11:slot 2
    uint8_t OutboundSlot;      //0:slot 11:slot 2
    uint8_t ChannelMode;        //0: direct connectionmode 4: true dual slot
    uint8_t EncryptSw;          //encryption switch1:enable 2:disable
    uint8_t EncryptKey[8];       //secret key
    uint8_t pwrsave;            //power save switch 2:disable 1:enable
    uint8_t volume;             //Volume 1-9
    uint8_t mic;                //micgain 0-5
    uint8_t relay;              //relay disconnectnet 2:disable 1:enable
    uint8_t tail;                //trailing character
};
```

Channel mode ska vara = 0?

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x22
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16 bit checksum value)
6, 7	LEN	2	data length	Digital structure length
0	DATA	1	data information	根据需要填 write digital structure , structure 后面的注释为变量的范围
	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x22
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x23: Set analog channel

```
tstruct db_analog_info
{
    uint8_t head;           // header 0x68
    uint8_t CMD;            // command 0x22
    uint8_t RW;              // operation type
    uint8_t SR;              // set/response
    uint16_t CKSUM;          // checksum
    uint8_t LEN_dummy;        // måste vara = 0
    uint8_t LEN;              // DATA length

    uint32_t rx_freq;         // receive frequency 400000000-48000000HZ
    uint32_t tx_freq;         // transmit frequency 400000000-48000000HZ
    uint8_t band;             // 0:narrow 1:wide
    uint8_t power;             // 0:low power 1:high power
    uint8_t sq;                // SQ level 0~9
    uint8_t rx_type;           // |0:carrier|1:ctesss |2:normal DCS|3:inverse DCS
    uint8_t rx_subcode;        // |0| 0~50 | 0~82 | 0~82
    uint8_t tx_type;           // |0:carrier |1:ctesss |2:normal DCS|3:inverse DCS
    uint8_t tx_subcode;        // |0| 0~50 | 0~82 | 0~82
    uint8_t pwrsave;           // power save switch 2:disable 1:enable
    uint8_t volume;             // Volume 1-9
    uint8_t monitor;            // monitor switch 2:disable 1:enable
    uint8_t relay;              // relay disconnect net 2:disable 1:enable
    uint8_t tail;                // trailing character
};
```

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x23
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	analog structure length
0	DATA	1	data information	Fill the analog structure , behind the structure is the comment of the variate range.
	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	lHead	1	start package	0x68
1	CMD	1	command	0x23
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x24: query digital channel information

Samma layout på query som t.ex. 0x32. Men anpassat svar motsvarande 0x22.

0x25: query analog channel information

Samma layout på query som t.ex. 0x32. Men anpassat svar motsvarande 0x23.

0x26: Set transmit information

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x26
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00 0x01
0	DATA	1	data information	1: send 开始 2: send 结束
	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x26
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00.0x00
8	TAIL	1	end package	0x10

0x27: query if the module finished the initialization

Samma layout på query som t.ex. 0x32. Men anpassat svar, verkar vara status 0 eller 1.

Anrop i samband med strömpåslag, för att vara säker på att DMR modulen är aktiverad.

0x28: Set enhanced function

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x28
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x05
8	FUN	1	enhanced function	0x01: walkie talkie detect 0x02: Call Prompting 0x03: remote monitoring 0x04: remote walkie talkie to be dead 0x05: walkie talkie activate
9,10,11,12	CallNum	4	contact number	1-16776415
12	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x29
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success
				0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x29: Set encryption function (inte tillåtet för amatörbruk)

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x29
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x09
8	SWITCH	1	encryption switch	1:enable 2:disable
9~16	DATA	8	data	encrypt 0~7 byte
17	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x29
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x2A: Set MICgain

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2A
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	GAIN	1	gain	Range :0~15
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2A
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x2B: Query digital voice receive information

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2B
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8		1		01
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2B
2	R/W	1	operation type	1
3	S/R	1	Set / reponse	1
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x05
8-12	DATA		data information	byte 0 is call type : 0x00: individual call 0x01: group call 0x02: no address call 0x03: call all and broadcast BYTE 1~4 : contact number
13	TAIL	1	end package	0x10

0x2C: send SMS

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2C
2	R/W	1	operation type	0x01
3	S/R	1	Set / reponse	0x01
4, 5	CKSUM	2	checksum	(16bit checksum ;)
6, 7	LEN	LEN	data length	
8	Msg Type	1	SMS type	0x01: IP with confirm 0x02: IP without confirm 0x03: group call
9,10,11,12	CallNum	4	contact number	contact number 4 byte
13	DATA	LEN-4	data information	SMS coding: Unicode
	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2C
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success
				0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

commnet: SMS forwardly send command

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2D
2	R/W	1	operation type	0x02 (forwardly send)
3	S/R	1	Set / reponse	0x01
4, 5	CKSUM	2	checksum	16bit checksum
6, 7	LEN	2	data length	
8-11	CallID	4	SMS sending part number	
12	MsgData	LEN-4	SMS content	SMS coding: Unicode
	TAIL	1	end package	0x10

0x2D: get SMS

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2D

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2D
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x01
4, 5	CKSUM	2	checksum	16bit checksum
6, 7	LEN	2	data length	
8-11	CallID	4	The number of SMS sending part	
12	MsgData	LEN-4	SMS content	SMS coding : Unicode
	TAIL	1	end package	0x10

2020-12-19

0x2E: Set Volume

Offset	Flag	Length	Comment	Detail
0	IHead	1	start package	0x68
1	CMD	1	command	0x2E
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	VOL	1	Volume	Range: 1~9, 9: max volume
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2E
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success
				0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x2F: Set monitor

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2F
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	VOL	1		1: enable 2:disable
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x2F
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x30: Set squelch (analogt)

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x31
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / reponse	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	VOL	1		1: enable 2:disable
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x31
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / reponse	0x00 Set success
				0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x31: Set power save mode

				0x01Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x32: query signal strength

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x32
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / response	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	Data	1	data information	0x01
	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x32
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / response	0x00 success 0x01 failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	
8	RSSI	1	data information	signal strength 值
	TAIL	1	end package	0x10

0x33: Set relay disconnect net mode

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x33
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / response	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	VOL	1		1: enable 2:disable
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x33
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / response	0x00 Set success 0x01 Set failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x00
8	TAIL	1	end package	0x10

0x34: Query version

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x34
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	Set / response	0x01: Set
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x01
8	DATA	1	data	0x01
9	TAIL	1	end package	0x10

response package

Offset	Flag	Length	Comment	Detail
0	Head	1	start package	0x68
1	CMD	1	command	0x34
2	R/W	1	operation type	0x00 (read mode)
3	S/R	1	Set / response	0x00 success 0x01 failure 0x02 verify wrong
4, 5	CKSUM	2	checksum	(16bit checksum)
6, 7	LEN	2	data length	0x00,0x12
8-25	DATA	18	软件 version	0~17 byte : 软件 version
26	TAIL	1	end package	0x10

0x3C: Pause transmit function

Offset	Flag	Length	Comment	Detail
0	Head	1	Start package	0x68
1	CMD	1	command	0x3C
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	set/response	0x01: Set
4,5	CKSUM	2	checksum	16 bit checksum
6,7	LEN	2	data length	0x00,0x01
8		1		1:enable 2:disable
9	TAiL	1	end package	0x10

Response Package

Offset	Flag	Length	Comment	Detail
0	Head	1	Start package	0x68
1	CMD	1	command	0x3C
2	R/W	1	operation type	0x00 (Read mode)
3	S/R	1	set/response	0x00: Set success 0x01: Set failure 0x02: Verify wrong
4,5	CKSUM	2	checksum	16 bit checksum
6,7	LEN	2	data length	0x00,0x00
9	TAiL	1	end package	0x10

0x3D: Get message

This is a message sent from the Host when a valid DMR voice message transmission begin. A similar message is sent when the transmission end.

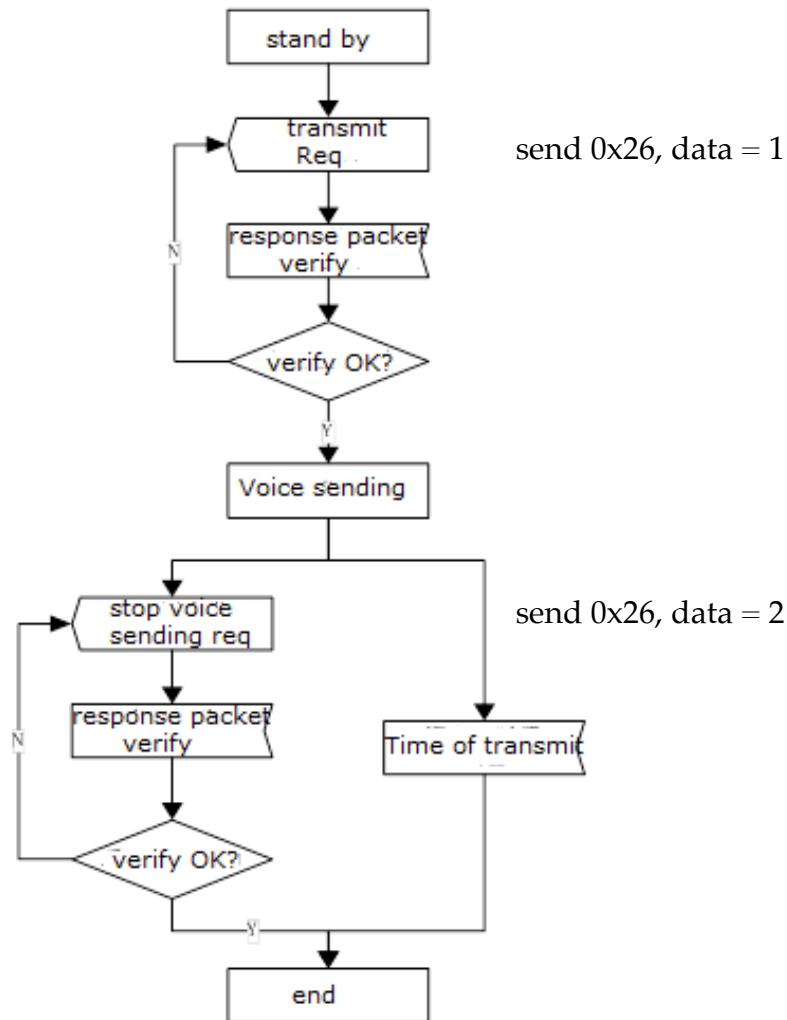
Offset	Flag	Length	Comment	Detail
0	Head	1	Start package	0x68
1	CMD	1	command	0x3D
2	R/W	1	operation type	0x01 (write mode)
3	S/R	1	operation type	0x01
4,5	CKSUM	2	checksum	16 bit checksum
6,7	LEN	2	data length	0x00,0x01
8		1		0x00 start of transmission 0x01 end of transmission
9	TAiL	1	end package	0x10

Voice transmission and receive

Voice transmission

For the convenience of users, you can configure the CALL pin to send voice (this function can also be used through the serial protocol to fulfill). When using PTT pin control, the operation goes as follows:

- Use the serial port command to write the channel switching configuration package to switch to the desired channel.
- CALL configuration as shown in the figure, pull CALL low to start transmission; pull CALL pin high to end transmission.



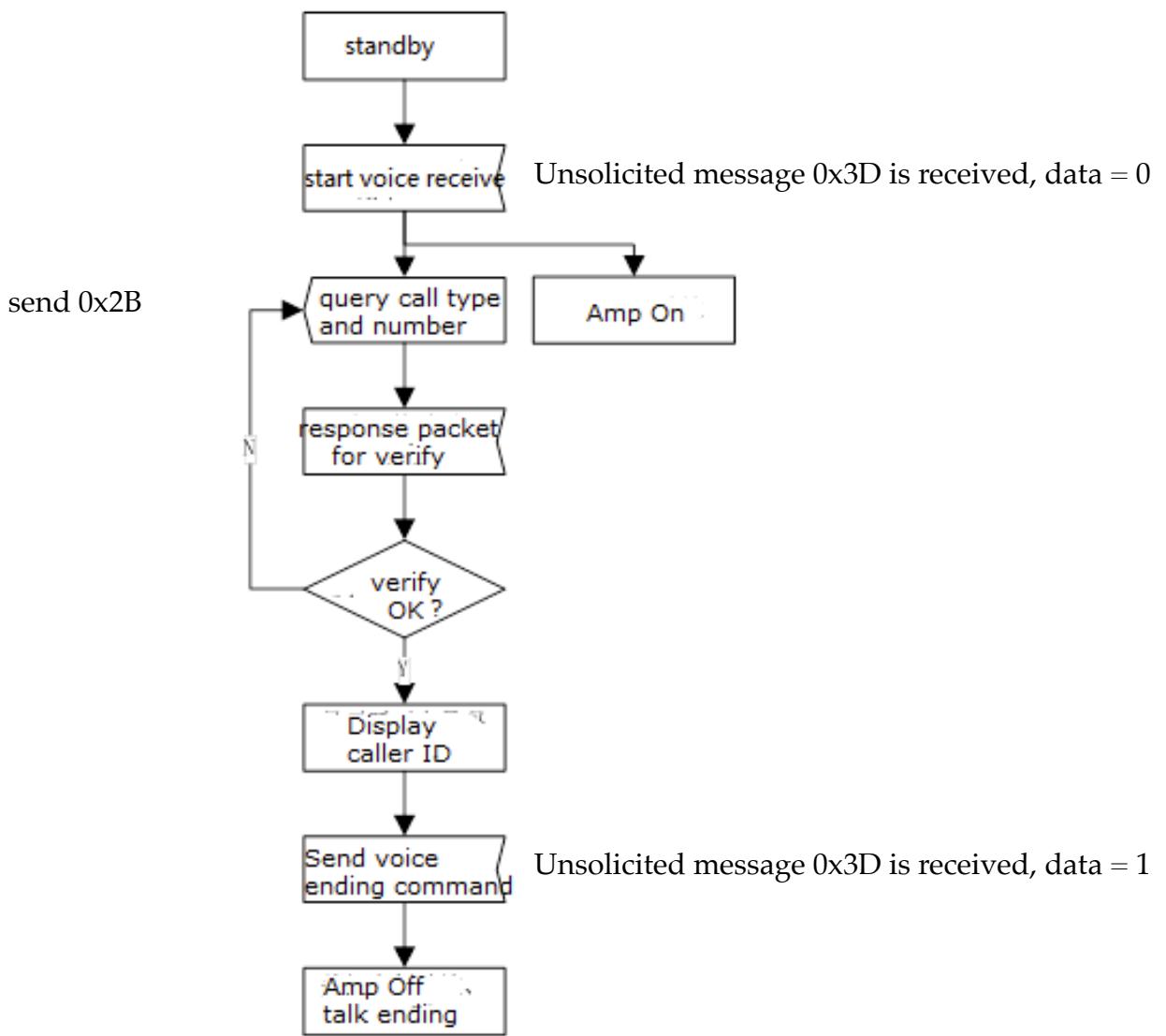
- 1) MCU send command "Start transmitting voice"
- 2) Module send the verify result (OK,or NG) Acknowledgeto HOST ;

3) If the verify OK, it means the module is transmitting the voice;
If verify NG, HOST send command“Start transmitting voice”again;

4) During the voice transmitting , MCU send command“Stop transmitting voice”

If the verify OK, it means the module voice transmitting is stopped;
If verify NG, HOST send command“Stop transmitting voice”again;
If module always can't get the “stop transmitting “command, the module will automatically stop transmit voice when reach the limited transmit time

Voice receive flowchart



- 1) Once module received voice, it send “Start voice receive” command to HOST; HOST get the voice receive request , then power on Amplifier, and query the calling type and number from module;
- 2) Module check the query command , if verify OK, module send the calling type and number to HOST, if verify NG, then send the “verify NG”to HOST.
- 3) HOST get the Acknowledgement from module, then check if it is OK, if OK, then display the calling number; if NG, query again.

4) If voice transmitting finished, module send the "voice transmitting stopped" information to HOST, HOST power off the amplifier. And display "voice stopped"

Short message communication (SMS)

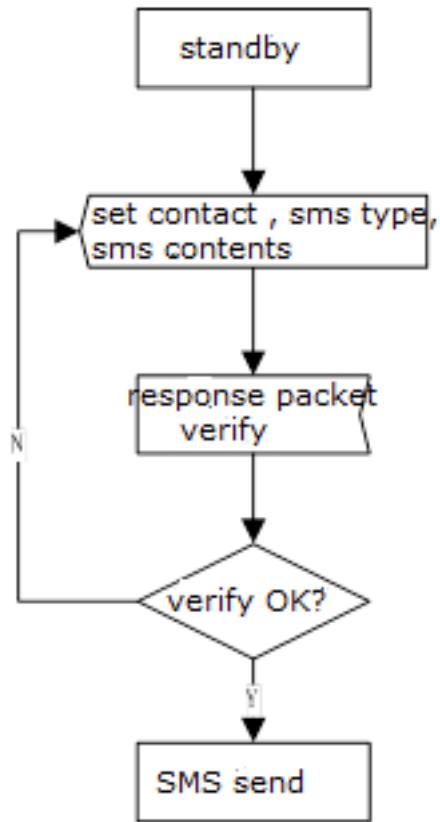
SMS main type is UnacknowledgedSMS, AcknowledgedSMS, and status SMS;

UnacknowledgedSMS usually used in the case of broadcast;

Acknowledged SMS usually used in the case of point –to-point; it has the ack function, is suit to reliable SMS sending;

Status SMS usually used in the case of scheduled command communication to improve the efficiency;

UnacknowledgedSMS sending flowchart



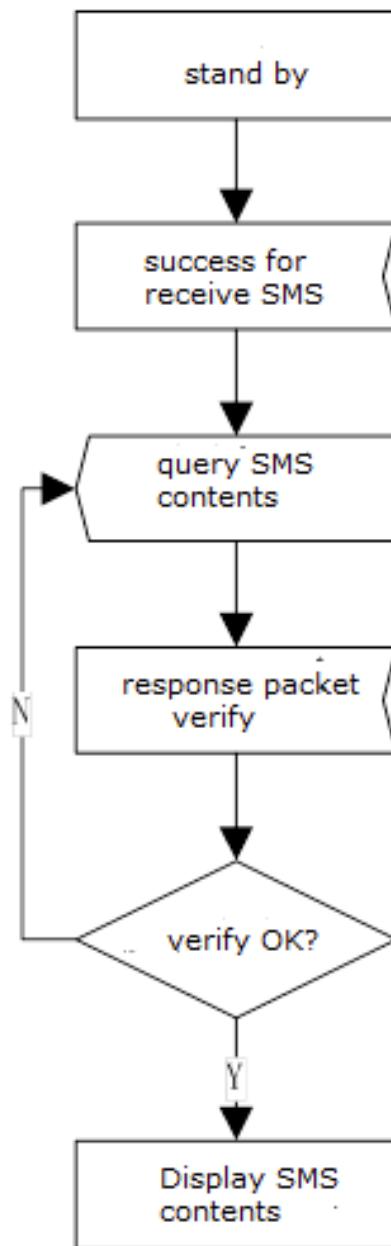
MCU send command to module vis UART port, SetSMS contact , Type, contents;

Module then verify the command, if verify OK, then send SMS;

If verify NG, then Acknowledgeto MCU with NG information;

Comment: if the module read the contact failure by contact index, it will Acknowledgeto MCUwith NG information;

UnacknowledgedSMS receive flowchart

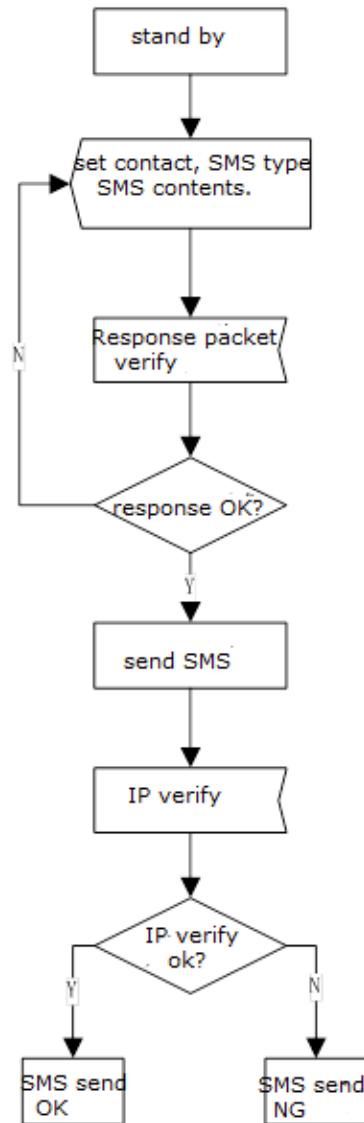


MC get the serial packet of SMS receive success from module, then send the command to query SMS contents;

Module verify the query command, if verify OK , then module send the SMS and caller ID to MCU;

If verify NG, Module send the NG information to MCU.

AcknowledgedSMS Sending flowchart



- 1) MCU send command to module for setting SMS contact, SMS mode, SMS contents;
- 2) Module verify the command, if verify OK, then send SMS;
If verify NG, then send NG information to MCU;
- 3) When finished the SMS sending, module send IP verify information to MCU that if the other side get the SMS or not.
- 4) MCU based on the IP Acknowledgeinformationto display the SMS sending OK nor NG.;

Comment: when module get the contact failure by contact index, it will send Acknowledgeto MCU with NG information;; When contact type is GROUP or General CALL , it will not judge

AcknowledgedSMS Receive flowchart

Same as UnacknowledgedSMS receive flowchart

MOTTAGET MEDDELANDE FRÅN ANYTONE

68
2D
02 forwardly send
00 response
F3 check
2F check

00 len
13 len

00 callID
E1 "
A0 "
24 "

00 header
0D "
00 "
0A "

00
41
00
44
00
47
00
4A
00
4D
00
4D
00
10

APPENDIX

UART checksum calculate

```
uint16 PcCheckSum(uint8 * buf, int16 len)
{
    uint32 sum=0;
    while(len >1)
    {
        sum += 0xFFFF & (*buf<<8|*(buf+1));
        buf+=2;
        len-=2;
    }
    if (len)
    {
        sum += (0xFF & *buf)<<8;
    }
    while (sum>>16)
    {
        sum = (sum & 0xFFFF)+(sum >> 16);
    }
    return( (uint16) sum ^ 0xFFFF);
}
```

I "Bilaga seriell verifieringsalgoritm" i dokumentet finns det två frågor om funktionen `PcCheckSum (uint8 * buf, int16 len):`

1. Parametern `buf`, är det startpositionen för startpaketet eller startpositionen för data, `len` är hela paketets längd eller datalängden?
2. Om det är verifieringssumman för det beräknade hela paketet, måste du ställa in fjärde och femte bitarna (verifieringssumman) till "0" innan du beräknar verifieringssumman?

```
uint8 CheckCkSum(uint16 len,uint8 buf[])
{
    uint16 sum,cksum;
    sum = buf[Pcksum];//Pcksum 为4
    buf[Pcksum] = 0;
    sum = (sum<<8);
    sum = sum + buf[Pcksum+1];
    buf[Pcksum+1] = 0;
    cksum = PcCheckSum(buf,len);
    buf[Pcksum+1] = (sum&0xff);
    buf[Pcksum] = (sum>>8)&0xff;
    if(sum == cksum)
    {
        //AckToPC(buf[Pcmd],ok);
        return 1;
    }
    else
    {
        //AckToPC(buf[Pcmd],ChkError);
        return 0;
    }
}
```

}

// Detta är testgränssnittet, du kan hänvisa till koden nedan.

Kan se av det

I "Bilaga seriell verifieringsalgoritm" i dokumentet finns det två frågor om funktionen
PcCheckSum (uint8 * buf, int16 len):

1buf startar från startpaketet, len är hela paketets längd

2. Ja, du måste ställa in fjärde och femte bitarna (positionen för verifieringssumman) till "0"
och sedan beräkna verifieringssumman

arduino kodexempel

<https://www.arduino.cc/reference/en/language/functions/communication/serial/>

<https://forum.arduino.cc/index.php?topic=396450>

Komplett apparat med enklare modul AT-kommando styrd

<https://www.instructables.com/id/Arduino-Walkie-Talkie/>

<http://www.kh-gps.de/dra.htm>

<http://ardutrx.generationmake.de/>

Arduino objekts

Channel

- 0x22: Set digital channel
- 0x23: Set analog channel
- 0x24: query digital channel information
- 0x25: query analog channel information

Transmit

- 0x26: Set transmit information
- 0x27: query if the module finished the initialization

Feature

- 0x28: Set enhanced function
- 0x29: Set encryption function

Call

- 0x2A: Set MICgain

Receive

- 0x2B: Query digital voice receive information

SMS

- 0x2C: send SMS
- 0x2D: get SMS

Receive

- 0x2E: Set Volume
- 0x2F: Set monitor
- 0x30: Set squelch

Module

- 0x31: Set power save mode

Transmit

- 0x32: Query signal strength
- 0x33: Set relay disconnect net mode

Module

0x34: Query version

Transmit

0x3C: Pause transmit function

Checksummeexempel

Offset	Flag	Length	Comment	Detail
0	Head	1	Head of the packet	0x68
1	CMD	1	Command	0x14
2	R/W	1	Operating Mode	0x01 (Write)
3	S/R	1	Set/reply Command	0x01: Set
4, 5	CKSUM	2	Checksum	(16bit)
6, 7	LEN	2	data length	0x00,0x03
8,9,10	Num	3	Group number	group call number
11	TAIL	1	End of the Packet	0x10

681401010000 0003 00ef33 10

8.9.7 software version set

Offset	Flag	Length	Comment	Detail
0	Head	1	Head of the packet	0x68
1	CMD	1	Command	0x16
2	R/W	1	Operating Mode	0x01 (Write)
3	S/R	1	Set/reply Command	0x01: Set
4, 5	CKSUM	2	Checksum	(16bit)
6, 7	LEN	2	data length	0x00,0x01
8,	FREQ	1		01
9	TAIL	1	End of the Packet	0x10

68 16 01 01 95 d7 00 01 01 10

Samlade erfarenheter

Testkörning mot DMR modul 202-09-08

Kommandon

Anrop	testat anropa	Information	
0x22: Set digital channel	x	Sätter upp en digital kanal. Förutsättning för att kunna lyssna och sända på DMR. Programmerad och fungerar.	CC =1 är det enda som fungerar just nu. Lovad får ny firmware 2020-10-11
0x23: Set analog channel	x	Motsvarande för analog kanal. Det senast sända 0x22 eller 0x23 gäller.	
0x24 Get digital channel information	x	Hämta DMR kanalens attribut. Enbart de 64 första tecknen tas emot pga begränsning i soft serial för Arduino Uno. Har ingen praktisk betydelse.	
0x25 get analog channel information	x	Motsvarande för analogt. Inget problem med 64 bytesgränsen då transaktionen är kort.	
0x26 set transmit information	x	Används för att starta sändning om data är enable resp stoppa om disable. Programmerad och fungerar.	
0x27 query if module finished the initialization	x	Vet egentligen inte, kanske för att kontrollera att modulen är uppstartat efter strömpåslag. Har lagt in test i setup.	
0x28 set enhanced function	--	Inte intressant just nu.	
0x29 set encryption function	--	Använder inte	
0x2A set mic gain	x	Programmerad, men inte testad	
0x2B query digital voice receive information	x	Lämnar information om aktuell mottagen station. DMRID ligger kvar efter samtalet tills nästa station tas emot.	
0x2C send SMS	x	Saknar info filler1 och filler2 + svenska tecken.	
0x2D get SMS	x	Programmerad, men går inte att verifiera	
0x2E set volume	x	Programmerad fungerar	
0x2F set monitor	x	Vet inte, har inte lyckats räkna ut vad den gör	
0x30 set squelch	x	Bara för analogt	
0x31 set power save mode	x	Modulen i strömsparläge. Inte verifierat.	
0x32 query signal strength	x	DMR: Används för att identifiera inkommande samtal, signalstyrka större än 0, Vital för receive loopen.	
0x33 set relay disconnect net mode	x	Vet inte vilken DMR funktion det motsvarar.	

0x34 query version	x	DMR modulens version. Programmerad i setup.	
0x3C pause transmit function	x	Vet inte, kanske låser sändfunktionen	
0x3D receive voice	x	DMR används för att identifiera när modulen tar emot röstsamtal. Programmerad i receiveslingan.	

Bandplan 432 - 438 MHz

Utgåva: 2019-11-28, SM7GVF

Bandplanen bygger på IARU Region 1 bandplan; IARU-R1 VHF Handbook del 2, sektion 1.5 (hittas antingen på IARU Region 1's hemsida eller i sektionens dokumentarkiv under IARU)

Länk till IARU: http://www.iaru-r1.org/index.php?option=com_content&view=article&id=191&Itemid=144

Alla fyra koordineras av IARU Region 1 Beacon Coordinator, genom SSA.



Frekvens MHz	Max bandbredd (-6 db)	Modulation	Användning
432,000 - 432,100	500 Hz	Telegrafi (a) MGM	432,050 Telegrafi aktivitetscenter
432,100 - 432,400	2700 Hz	Telegrafi SSB MGM	432,200 SSB aktivitetscenter 432,350 Mikrovågor talk back 432,370 MGM (FSK411) anropsfrekvens (MS)
432,400 - 432,490	500 Hz	Telegrafi/MGM	Fyrar exklusivt
432,491 - 432,493	500 Hz	EMGM	Experimentell MGM
432,5	12 kHz	Alla moder	432,500 APRS ny frekvens
			432,5125 RU361
432,5125 - 432,975	12 kHz	FM, DV	Repeater in 2 MHz skift 12,5 kHz kanalavstånd 432,975 RU398
433,000 - 433,375	12 kHz	FM, DV	(c)
433,400 - 433,575	12 kHz	FM, DV	433,400 SSTV FM/AFSK U272 433,500 FM anropsfrekvens U280 433,400 - 433,575 Simplex 12,5 KHz kanalavstånd U272-U286 433,450 DV anrop (d)
433,600 - 434,000		Alla moder	433,600 Data aktivitetscenter 433,625 433,775 Digitala kanaler U290-302 433,950 433,9625 433,9975 433,9875 FM simplex Internet gateway 433,775 LoRa-1, node->GW; 433,900 LoRa-2, GW->node (f)
434,000 - 434,500	12 kHz	FM, DV	434,0125 434,025 434,050 FM simplex Internet gateway 434,450 - 434,500 Digitala kanaler, DV internet gateway
434,5125 - 434,975	12 kHz	FM, DV	434,5125 RU361 Repeater utfrékvens 12,5 kHz kanalavstånd 434,975 RU398
435,000 - 438,000	20 kHz	Alla moder	Exklusivt satellitsegment (e)

Fotnoter

(a) Telegrafi är tillåtet över hela bandet

(b) I Norden har vi bara 432-438 MHz, i Europa är det 430-440 MHz. Användningen skiljer mellan dessa regioner varför hänsyn måste tas vid nyttjandet av dessa frekvenser.

(c) Tidigare repeater Input, nyttjas för FM och DV. Avvecklat 2004 då -2 MHz skift infördes.

(d) DV simplex, ej avsett för gateways.

(e) Sändning från DATV repeatrar skall inte förekomma i bandet 436 - 438 MHz.

(f) LORA med 125 kHz bandbredd

Alla fyra är koordinerade av IARU Region 1 beacon coordinator (via SSA fyrfunktionär).

Bandplanen bygger på IARU Region 1 bandplan.

Övriga Europa har 430-440 MHz, se skillnader i VHF Managers Handbook.

Bandplanen bygger på IARU Region 1 bandplan.

Anslutning till Arduino

IO pinnar

```

// const byte modePin = 6;      //input, switching between modes

// const byte squelchPin = 15;  //input, this is actually A1,
                           //but we are to use it for squelch button, SQL = 0

//these are all the pins, after PCB released, all pins are fixed, it is unnecessary keep using
name, changed to numbers

// const byte pinA = 3;        //input, digital pin 3, (pinB,pinA) clockwise turning as
                           //increase. swap them if you prefer counter clockwise

// const byte pinB = 2;        //input, digital pin 2, (pinB,pinA) both pins are pulled up,
                           //no need for resistors

// const byte scanPin = 4;     //input, center button of rotary sensor, pulled up HIGH to
                           //switch to/from scan mode, active LOW

// const byte PTTPin = 5;      //input, used to switch TX/RX, connect to RF module PTT
                           //pin & PTT button, active LOW

// const byte modePin = 6;      //input, switching between modes

// const byte backPin = 7;      //input, pin to go back from menu & jump back to normal
                           //receiving mode

// 8 = TX; 9 = RX           //used by rfSerial for communication with RF module

// const uint8_t CS_PIN = 10;   //CS enable pin for the Digital Potentiometer chip
                           //MCP41010, can be changed to anyone except pin 12;

// 11,12,13 SPI pins         //occupied by the SPI interface although 12 was NOT
                           //physically used, cannot make use of it

// const byte beepPin = 14;    //beepPin; OUTPUT, this is actually A0 but we are to use
                           //it for the beeper as pin 13 was used by SPI in this case

// const byte squelchPin = 15; //input, this is actually A1, but we are to use it for squelch
                           //button, SQL = 0

// const byte signalPin = 16;  //input, this is actually A2, detecting signal received from
                           //RF module, active LOW

// const byte btpowerPin = 17; //OUTPUT, this is actually A3, use it to power switch
                           //bluetooth module with an PNP transistor, active LOW

// A4 =18=SDA; A5 =19=SCL  //used by the IIC OLED display

// A6;                      //the only analog pins left

// const byte voltagePin = A7; //used on monitoring the voltage of the battery

```

Kanalnummer

UHF RU361- RU367 434,5125 – 434,5875 Digital trafik

UHF RU368 – RU398 434,600 – 434,975

(434,5125 RU361 Repeater utfrekvens 12,5 kHz kanalavstånd 434,975 RU398)

433,400 SSTV FM / AFSK U272

433,500 FM anropsfrekvens U280

433,400 - 433,575 Simplex 12,5 KHz kanalavstånd U272-U286

433,450 DV anrop (d)

433,600 Data aktivitetscenter

433,625-433,775 Digitala kanaler U290-302

433,950 433,9625 433,9975 433,9875 FM simplex Internet gateway

Kanal = 430 + 0,0125 * kanalnummer MHz

Termer för kanaler Anytone och DMR module

Anytone

Channel type

A Analog

D Digital

A+D TX A

D+A TX D

TX power

Lo/Mi/Hi/Turbo

Offset

+Hz

Band width

Narrow

Rx freq

432.58750

Tx freq

432.58750

TalkAround

off/on

Name

Unikt namn för channel

Tx Allow

Always

Channel Free

Different CC

Same CC

Tx Prohibit

off/on

Radio ID

SM7ECA

Color Code

0-

Time Slot

TS1/TS2

Digi Encrypt

off/encrypt 1-

Encrypt Type

Normal

Enhance

Rx Group List

Talk Groups

Work Alone

off/on

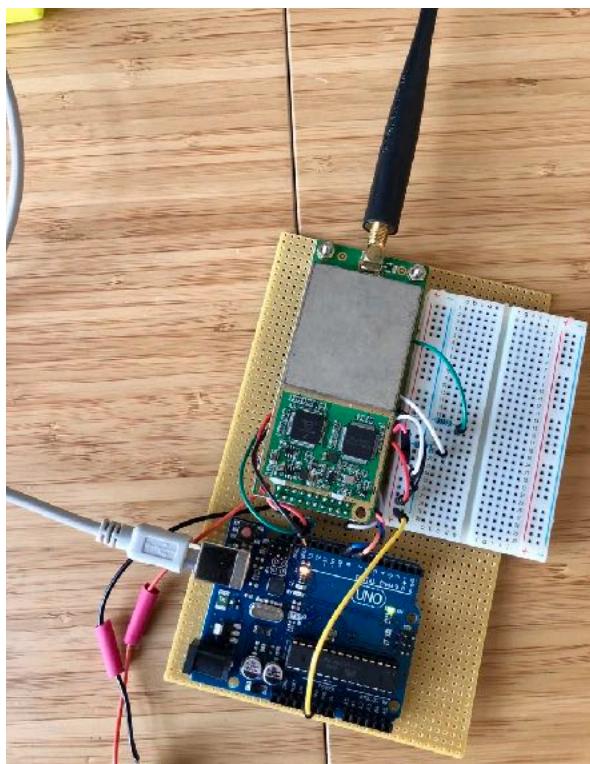
CH Ranging

off/on
 APRS Receive
 off/on
 SMS Forbid
 off/on
 DataAck Forbid
 off/on
 DMR Mode
 Simplex
 Repeater
 Double Slot

DMR modul		Anytone
rx_freq;	receive frequency 400000000-480000000HZ	Rx freq 432.00000 -
tx_freq;	transmit frequency 400000000-480000000HZ	Tx freq 432.00000 -
localID;	local ID1-16776415	DMR ID för ägaren
GroupList[32];	receive group list	TG lista som monitoreras
tx_contact;	contact number 1-16776415	Talk group jag sänder på
ContactType;	contact type 0:individual call 1:group call2:call all 3: full duplex	normalt 1?
power;	0:low power 1: high power	power
cc;	color code0~15	CC
InboundSlot;	0:slot 11:slot 2	Time Slot 1 eller 2
OutboundSlot;	0:slot 11:slot 2	alltid samma på in och ut
ChannelMode;	0: direct connection 4: true dual slot	TalkAround off om dual
EncryptSw;	encryption switch 1:enable 2:disable	Digi Encrypt disable
EncryptKey[8];	secret key	använder inte den
pwrsave;	power save switch 2:disable 1:enable	inte på channelnivå
volume;	Volume 1-9	- " -
mic;	micgain 0~2	- " -
relay;	relay disconnectnet 2:disable 1:enable	??

Projektlogg DMR enheten

2020-09-15



Sändning sker alltid på CC=1 oavsett vad man skickar cc-fältet till modulen.
rx_freq och tx_freq ska anges med 9 siffror som ovan, inte 10 som i manualen.

0x24: Arduino hinner inte ta emot all data i svaret, bara de 64 första tecknen sen är bufferten full. Går inte att öka med parameter till SerialSoftware .h-filen. Kastas nog bort at processorn i Arduino. Ger ett overflow-meddelande. Har egentligen ingen praktisk betydelse för min applikation.

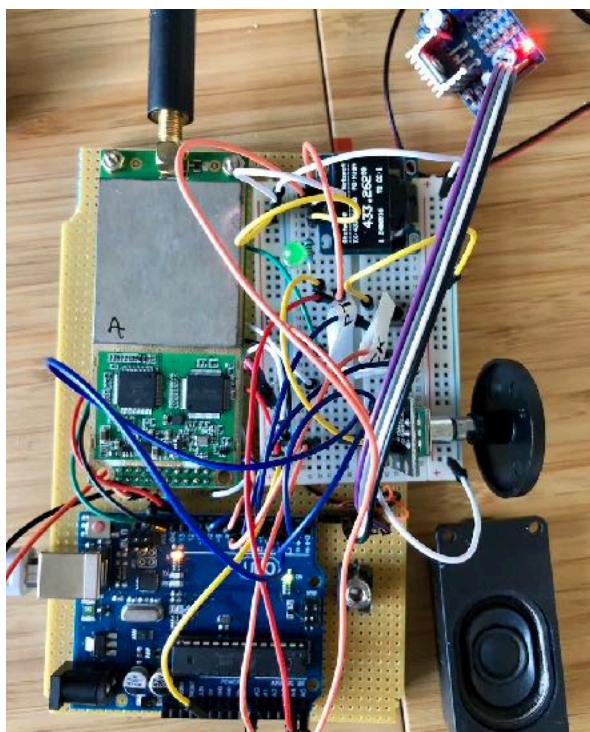
Nästa steg är att effektivisera koden. Mycket duplicerad kod i procedurerna.
Vänta på Reinos kort för styrning av radion.
Elektriskt.
Koppla in sladdar till mikrofon och förstärkare på DMR-kortet.

2020-09-18

SMS, testat sändning och konstaterat att meddelande kodas i UCS-2 dubbelbyte. Dvs 0x20 blir 0x0020 och lagrat i som 0x2000 i long int. Testat med åäö som inte verkar fungera mot Anytone. De sändtester som gjorts är direkt anslutet. Fortfarande problem vi Brandmeister.

Mottagning av SMS går knackigt. Tror inte jag fått det att fungera i något fall än.
Måste läsa på hur konvertering ska ske mellan integer och string.

2020-10-12



Implementerat "korrekt" receiveslinga med det odokumenterade kommandot 03D. Fick testprogram för PC av Frank där det syns tydligt vad som händer när man får inkommande samtal. Rotationsavkodare (RE) implementerad med enkel menufunktion. Inte komplett men fungerar tillräckligt bra för test. REs knapp används för att gå till meny och för att markera (kort klick) och backa upp från meny (lång klick). Inte helt perfekt. Ska testa med att byta till BACK knappen för att backa och MODE för att gå till meny. RE utan klick = volymkontroll.

Menypunkter

- Mic volume (0-15)
- Tx Talkgroup (240 ... 240240)
- ColorCode (0-15)
- TimeSlot (1,2)

Köpt och installerat en enkel förstärkarmodul för att kunna testa mottagning. Parallelkkör

med Anytone mot Hotspot och det fungerar perfekt just nu.

CC=1 är det enda som fungerar just nu. Rapporterat och verifierat. Utlovd att ny firmware att ladda ner kommer (i går).

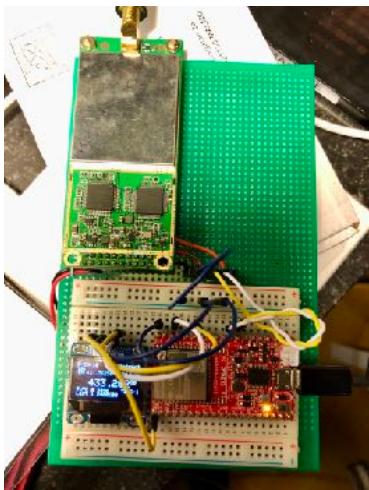
Har testat med fram till vad de sista parametrarna i inställning ska stå på för att fungera. ChannelMode = 0 (direct connection mode) öppnade mottagaren för lyssning. Innan dess tyst oavsett inställningar.

relay = 2 (relay disconnected net - disabled) just nu fungerar det. Vet inte vad som avses med parametern.

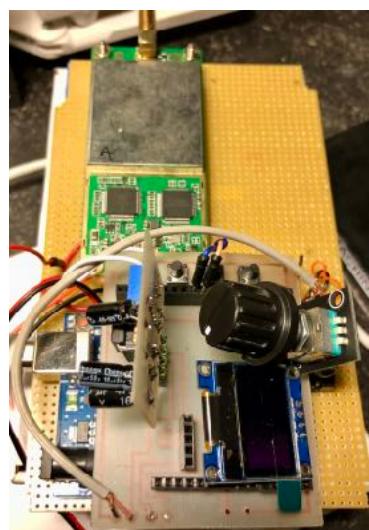
Har ett separat DMR ID för modulen, 2400530.

Inget test med SMS sedan 18/9.

2020-10-24



Konverterat programvaran till ESP32miljö för att få snabbare system, mer primärminne, inbyggt WIFI, Bluetooth. Den har också tre st UART, vilken gör att kommunikation med DMR fungerar bättre. Inga begränsningar på buffertstorlek till 64 kb som varit problem för ett av kommandona. Vald modul är något större än Arduino Nano men mindre än Uno och innebär att det är enklare bygga in modulen i den svarta låda som Reino menar att vi kan använda.



Innan byte av processor hann vi verifiera funktionaliteten. Har testat sändning och det är känsligt med störningar på mikrofoningången. Lyckades inte med kondensatorelementet som är det vanliga idag. Av misstag kopplade jag in det lilla högtalarelementet i monologen. Det fungerade nästa som det var tänkt. Körde första QSO med SA6OPT via Hotspoten. Hörde min egen signal på Hallandsåsrepeatern så det är en fungerande lösning. Fortfarande ingen lösning på CC problemet så vi gjorde ett test med att koppla om SA6OPTs repeater med CC=1 och det fungerade. Körde Parrot och signalen gick igenom och tillbaka (innan jag kopplat in mikrofon). Har fått testmodulen med audio och knappar från Reino och allt fungerade efter lite test. Den oledmodul Reino hade köpt var kopplad annorlunda, men inget problem efter omkoppling i programvaran. Audiomodulen också OK efter att jag vänt IC-kretsen om ;).

Då var DMR voice på banan med menysystem för att ställa in volym, mik-volym, tx-talgrupp, CC och TimeSlot med mekaniska komponenter.



Testat med att placera DMR-modulen på kant i boxen. Diskuterat med Reino om hur audiodelen ska skärmas för att inte ta in störningar från sändaren. Blir en modulen på baksidan av det kort som DMR-modulen monteras på. Efter lite funderande beslöt jag att eliminera rörliga delar och att i stället använda Nextion touchskärm för HMI. Beställt två 3,2 tums enheter från Bangpong.

Har just verifierat att man kan använda wifi för att koppla upp sig mot radioid's DMRid databas.Verifierat kommunikationen med DMR modulen vid samtalshantering. Modulen sänder ett meddelande vid start voice och ett vid avslut. Fångar upp och läser upp vilken DMR id som talar och läser in kompletterande info från radioid.net via deras web-API. Har nu basen för att bygga ett bra HMI.

```
start 3xD 3132236
{"count":1,"results":[{"callsign":"W7GO","city":"Las Vegas","country":"United States","fname":"Anthony","id":3132236,"remarks":"Portable","state":"Nevada","surname":"
end 3xD
start 3xD 3170059
{"count":1,"results":[{"callsign":"KE8PIP","city":"Connecut","country":"United States","fname":"Ryan","id":3170059,"remarks":"DMR","state":"Onic","surname":"Frdlic"}]
end 3xD
start 3xD 3170437
{"count":1,"results":[{"callsign":"KD2JPP","city":"Memphis","country":"United States","fname":"Robert","id":3170437,"remarks":"DMR","state":"New York","surname":"Hak"}]
end 3xD
start 3xD 3170612
{"count":1,"results":[{"callsign":"KD2JQX","city":"River Vale","country":"United States","fname":"Edmund","id":3170612,"remarks":"DMR","state":"New Jersey","surname":"
end 3xD
start 3xD 3163398
{"count":1,"results":[{"callsign":"KK4ZPY","city":"Louisville","country":"United States","fname":"Christopher","id":3163398,"remarks":"DMR","state":"Kentucky","surname":"
end 3xD
start 3xD 3160722
{"count":1,"results":[{"callsign":"WA5VUC","city":"Belfair","country":"United States","fname":"Stanley D","id":3160722,"remarks":"DMR","state":"Washington","surname":"
end 3xD
start 3xD 2190238
{"count":1,"results":[{"callsign":"9A2M","city":"Mirkovci","country":"Croatia","fname":"Hrvoje","id":2190238,"state":""}]}
end 3xD
start 3xD 2930092
{"count":1,"results":[{"callsign":"5S2RS","city":"Smarjeske Toplice","country":"Slovenia","fname":"","id":2930092,"state":"SI"}]}
end 3xD
start 3xD 3126837
{"count":1,"results":[{"callsign":"KB8FK","city":"Lake Orion","country":"United States","fname":"Scott","id":3126837,"remarks":"DMR","state":"Michigan","surname":"Lu"}]
end 3xD
```

2020-10-27

Firmware uppdatering från Sunrise löste CC problemet. Har nu också testkört parrot mot R7RJL i Lund både i låg effekt och högeffektläge. Funkar fint.

SK7RJL						<input checked="" type="checkbox"/> Last Heard (TG Filter)	<input checked="" type="checkbox"/> Senast hörd
Tid	Master	Anropssignal	Destination	Flaggor	RSSI		
18 Seconds	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S8		
24 Minutes	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S9+10dB		
24 Minutes	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S9+10dB		

Felsökte längre och letade fel i parameteruppsättningen. Efter att ha förklarat mitt problem för Peter som hittade jag felet. En skarvkontakt på antennkabeln hade skruvat upp sig och gjorde att signalen inte gick ut ordentligt. Kostade några timmar i går.

Förresten så fick jag först en version av firmware som inte fungerade. Först hade jag problem att överhuvudtaget ladda in i modulen. Två fel, del hade de angivit fel i vilka parametrar som skulle ställas in i terminalprogrammet (skulle vara none i sista parametern) som jag hittade efter mycket test och funderande. Sedan var programnamnet för långt för att accepteras av ymodem-programmet. Ändrade namnet till ett kortare .. och sedan gick det att ladda in. Tyvärr startade inte modulen med ny firmware. Skypade och frågade om de verkligen verifierat att det fungerar. Tydligen inte för efter några timmar fick jag en ny fungerande version. Mina moduler hade en annan hårdvarukonfiguration som inte var kompatibel med den firmware de sätta till att börja med. Dvs 3 fel, .. sedan rätt.

Undersökning av vilka parameterkombinationer som fungerar mot pi-star

Tid	Master	Anropssignal	Destination	Flaggor	RSSI
33 Seconds	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S9
2 Minutes	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S8
4 Minutes	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S9
6 Minutes	2401	SM7ECA [Arne] (2400530)	Parrot (9990)	TS1 DMR	★ S8

ConnectMode	0	4	0	4
Relay	2	2	1	2
35s				s9
2 min				s8
4 min				s9
6 min				s8

Ska man dra slutsatsen att ConnectMode = 4 är korrekt/bäst? Så är det säkert.

ChannelMode; //0: direct connectionmode 4: true dual slot (DMR tier 2)

Vad gör relay parametern ??? Sätter den till disable tills vidare.

relay; //relay disconnect net 2:disable 1:enable

2020-10-30

Falkenberg, kopplat upp mej mot SA6OPT repeater och konstaterat man måste scanna båda tidsluckorna om man vill höra allt. Lade in logik som kopplar om mellan tidsluckorna var 5e sekund. Det verkar fungera som tänkt. Har också finslipat på logiken för att hämta DM Rid information från radiooid.net. Lade in ett json-bibliotek med funktioner för att maska fram de enskilda fälten.

Gått igenom en praktisk lösning för montering i boxen och kommit överens om vad som ska göras med Reino.

Micael har 3D printa masker för displayerna. Ska testas i slutet av nästa vecka.

Olöst är fortfarande:

- uppdateringsrutin för parametrar eget DM Rid och call, WLANid och password. Kan kanske göras via usb eller Bluetooth som i Arduino Walkie Talkie.
- smart rutin för att spara undan senast funna DM Rid i lista för att slippa vara uppkopplad hela tiden. Måste vi montera en sd kortsläsare för att spara undan den? Gäller förmodligen också aktuella repeatrar och frekvenser.
- montering av display så den kan vinklas.
- kondensatormikrofon lösning med 3.5 mm kontakt.

2020-12-19

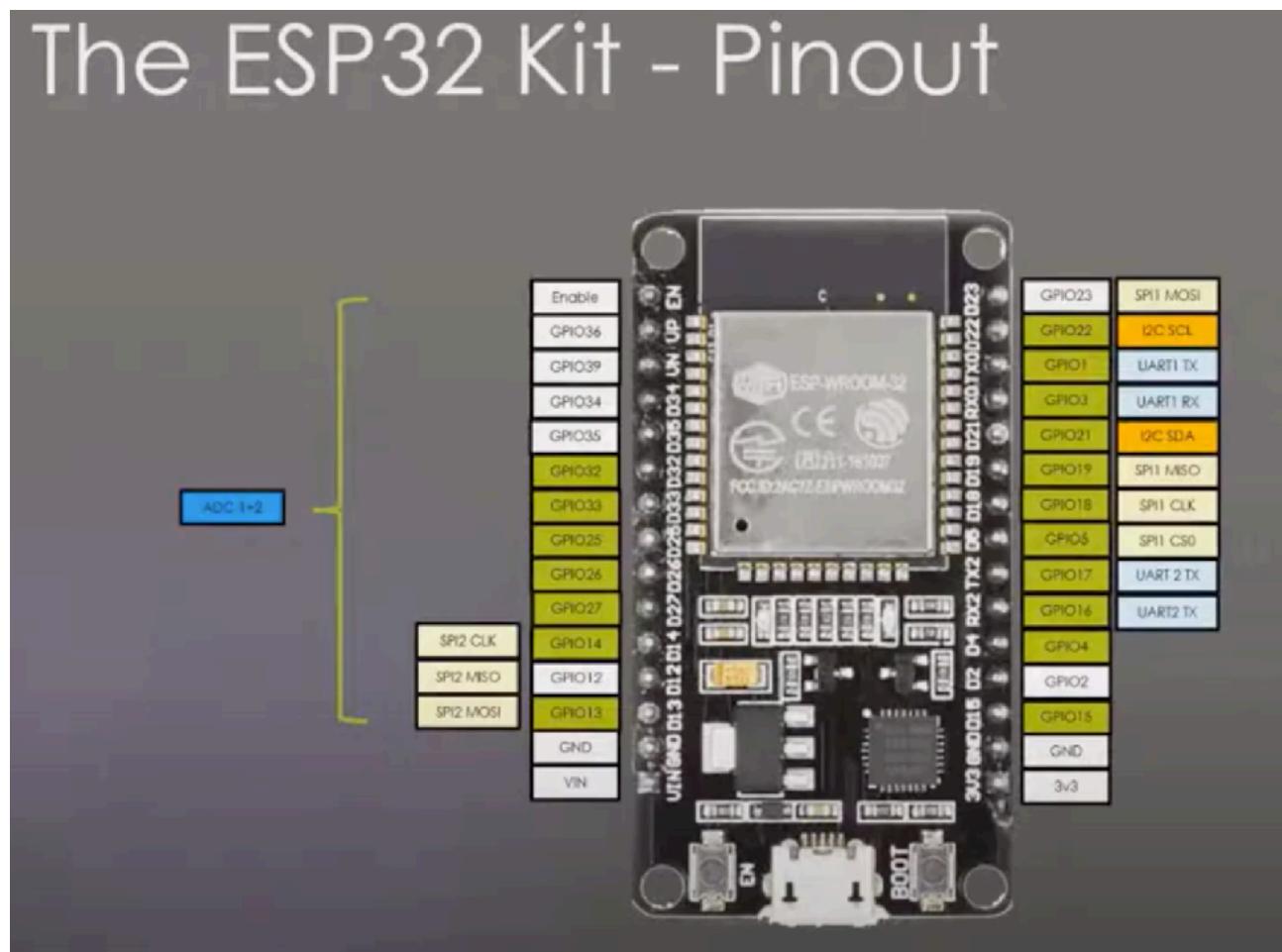
Bytt till Nextion under november.

	pin		
Serieportar	CPU	Nextion	DMRmodul
RXD1	2	TX	
TXD1	4	RX	
RXD2	16		TX
TXD2	17		RX

Planerar för installation av SD-kortläsare.

	pin	ansl	pin
SD-kort	CPU		SD-kort
CS	GPIO5	H10	—
SCK	GPIO18	H11	—
MOSI	GPIO23	H18	—
MISO	GPIO19	H12	—
VCC	5V	V1	—
GND	Gnd	H19	—
			H25

H= höger sida, V=vänster sida



Tabeller som behöver hanteras internt

Receive group

En lista av talgrupper som är öppna för att lyssna på för en channel. Flera listor kan läggas upp. Adderas i GroupList när en channel uppdateras med 0x22.

Lista	TGID	Namn	Contact Type
-------	------	------	--------------

Channel list

Lista över kanaler som är valbara via vredet på apparaten. Läggs in i EEPROM. Kommer troligen inte se ut så här.

ChannelNo		kanalnummer 1-40
Name		kanalnamn som visas i display
rx_freq;	receive frequency 400000000-480000000HZ	
tx_freq;	transmit frequency 400000000-480000000HZ	
localID;	local ID1-16776415	DMR ID för ägaren
Group;	receive talkgroup list	TG lista som monitoreras
tx_contact;	contact number 1-16776415	Se receive group ovan
ContactType;	contact type 0:individual call 1:group call2:call all 3: full duplex	Talk group jag sänder på normalt 1
power;	0:low power 1: high power	power
cc;	color code0~15	CC
Timeslot ;	0:slot 11:slot 2	inbound och outbound lika
ChannelMode;	0: direct connection 4: true dual slot	0 eller 4

DMRid från radioid.net

DMRid
call
name
city
state

Repeaterlista

DMRid
call
geografisk placering
tabell med TS + TG

Hotspot konfiguration

General Configuration	
Setting	Value
Hostname:	jullen11 <small>Do not add suffixes such as .local</small>
Node Callsign:	SM7ECA
CCS7/DMR ID:	2400481
Radio Frequency:	433.262.500 MHz
Latitude:	56.8897 degrees (positive value for North, negative for South)
Longitude:	12.4987 degrees (positive value for East, negative for West)
Town:	Falkenberg
Country:	Sweden
URL:	http://www.mw0mwz.co.uk/pi-star/ <input type="radio"/> Auto <input checked="" type="radio"/> Manual
Radio/Modem Type:	STM32-DVM / MMDVM_HS - Raspberry Pi Hat (GPIO) <input type="button" value="▼"/>
Node Type:	<input type="radio"/> Private <input checked="" type="radio"/> Public
APRS Host:	sweden.aprs2.net <input type="button" value="▼"/>
System Time Zone:	Europe/Stockholm <input type="button" value="▼"/>
Dashboard Language:	english_uk <input type="button" value="▼"/>
<input type="button" value="Apply Changes"/>	

DMR Configuration	
Setting	Value
DMR Master:	BM_Sweden_2401 <input type="button" value="▼"/>
Hotspot Security:	*****
BrandMeister Network:	Repeater Information Edit Repeater (BrandMeister Selfcare)
DMR ESSID:	2400481_02 <input type="button" value="▼"/>
DMR Colour Code:	1 <input type="button" value="▼"/>
DMR EmbeddedLCOnly:	<input type="checkbox"/>
DMR DumpTAData:	<input checked="" type="checkbox"/>
<input type="button" value="Apply Changes"/>	

Ändrade Node Type till public för att kunna använda 2400530 på Hotspoten.