DDS_Ctrl control program for Arduino Mega 2560.

Introduction.

The proposed control program is designed to operate on the Arduino Mega 2560 platform or any compatible board.
The program provides control of the DDS-synthesizer RN3AUS with any version of the firmware, including the firmware offered by the GM radio amateur. In addition, control of an AD9834-based synthesizer is supported.

Both LCD-modules ("LCD KeyPad-shield for Arduino") and OLED display with I2C interface can be used.
Control using the simplest five-button keyboard connected to the A0 input of the Arduino board. This keyboard is included in the LCD shield. Otherwise, it will not be difficult to make such a keyboard yourself according to the diagram below.

The program provides:

- Transmit in modes: CW, QRSS, DFCW, HELL, OP-32, WSPR-2 and-15, JT-9, FST-4 (W)
- Individual setting of operating frequencies for each mode
- Entering messages from the keyboard for each of the modes
- Programming of the "cyclogram" of the automatic beacon, consisting of transmissions in any modes, up to 15 transmissions in a series.
- Setting the exact time both manually and using a plug-in GPS-module NEO-6M or similar. This provides automatic detection of the QTH locator. When an additional RTC-module based on DS1307 with I2C interface is connected, the exact time is taken from it in the absence of GPS data.
- The ability to connect a computer with the DDS_Ctrl.exe program for direct control of the synthesizer, which is useful when working in other modes that are not implemented in the Arduino control program.
- Providing a direct connection to the GPS module, while on the computer you can use the program (U-center) to configure this module and obtain detailed navigation information.
- In all cases, communication with the computer is implemented in the standard way for Arduino - via a USB port.

Installing the required software and uploading the sketch to Arduino.

To download the program ("sketch") in Arduino, you need to install the Arduino development environment on your computer, which can be downloaded at: https://www.arduino.cc/en/software

Note: When using Windows XP, it may be difficult to install the USB port driver. On versions starting from Win7, the installation proceeds without problems.

Next, perform the installation process in a folder, for example, c:\Program Files\Arduino\

Unpack the attached libraries RTClib.zip and OLED_I2C.zip to c: \ Program Files \ Arduino \ libraries \. Attention! Please use the included OLED_I2C library as it has been patched to ensure it works in real time. The standard library of the same name, available on the Internet, will work with a large lag in the Arduino system clock!

Copy the sketch folder to your hard drive, for example c:\Radio\DDS_Ctrl_Arduino\myDDS_Ctrl\

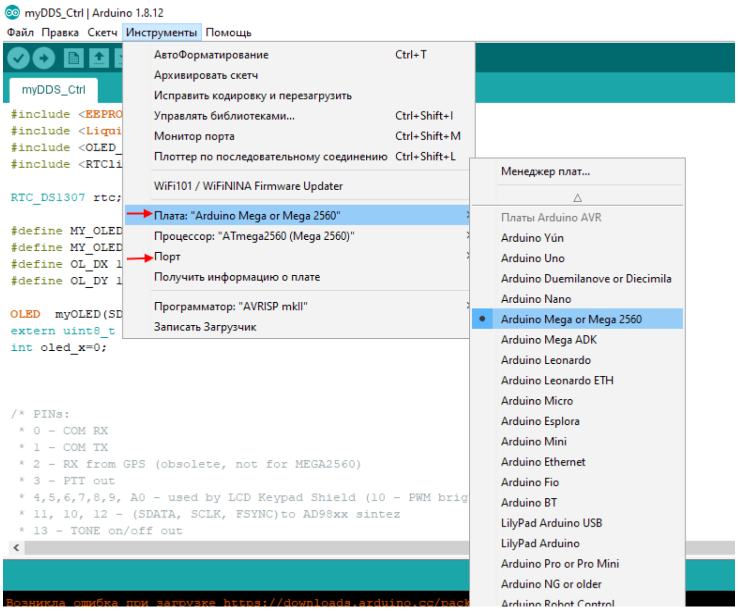
Start development environment.
Upload sketch: File-Open-C: \ Radio \ DDS_Ctrl_Arduino \ myDDS_Ctrl \ myDDS_Ctrl.ino

Connect the Arduino board to the USB port of your computer.

Select port: Tools-Port-COM4 - the port number may be different in your system.
Select Tools-Board-Arduino Mega 2560.

Start compiling and uploading the sketch by clicking the "Upload" button (Menu-Sketch-Upload)

The process will take a couple of minutes and should complete successfully. In this case, several warnings concerning the syntax of the language will be displayed in the compiler's message area; these messages can be ignored.


myDDS_Ctrl | Arduino 1.8.12
Файл Правка Скетч Инструменты Помощь
myDDS_Ctrl
АвтоФорматирование Ctrl+T
Архивировать скетч
Исправить кодировку и перезагрузить
Управлять библиотеками... Ctrl+Shift+I
Монитор порта Ctrl+Shift+M
Плоттер по последовательному соединению Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Плата: "Arduino Mega or Mega 2560"
Процессор: "ATmega2560 (Mega 2560)"
Порт
Получить информацию о плате
Программатор: "AVRISP mkII"
Записать Загрузчик
Менеджер плат...
△
Платы Arduino AVR
Arduino Yún
Arduino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
● Arduino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control

```
#include <EEPRO
#include <Liqui
#include <OLED_
#include <RTCli

RTC_DS1307 rtc;

#define MY_OLED
#define MY_OLED
#define OL_DX 1
#define OL_DY 1

OLED  myOLED(SD
extern uint8_t
int oled_x=0;



/* PINs:
 * 0 - COM RX
 * 1 - COM TX
 * 2 - RX from GPS (obsolete, not for MEGA2560)
 * 3 - PTT out
 * 4,5,6,7,8,9, A0 - used by LCD Keypad Shield (10 - PWM brig
 * 11, 10, 12 - (SDATA, SCLK, FSYNC)to AD98xx sintez
 * 13 - TONE on/off out
```

```
Возникла ошибка при загрузке https://downloads.arduino.cc/pack
```

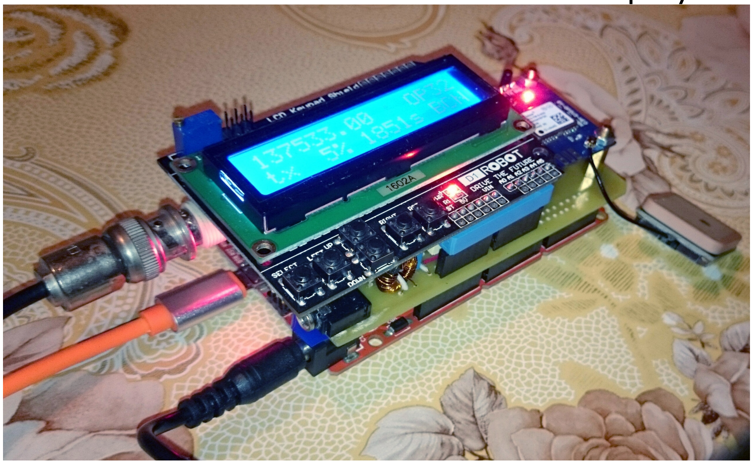
This completes the preparation of the Arduino board for work.

Connection to synthesizer and auxiliary modules.

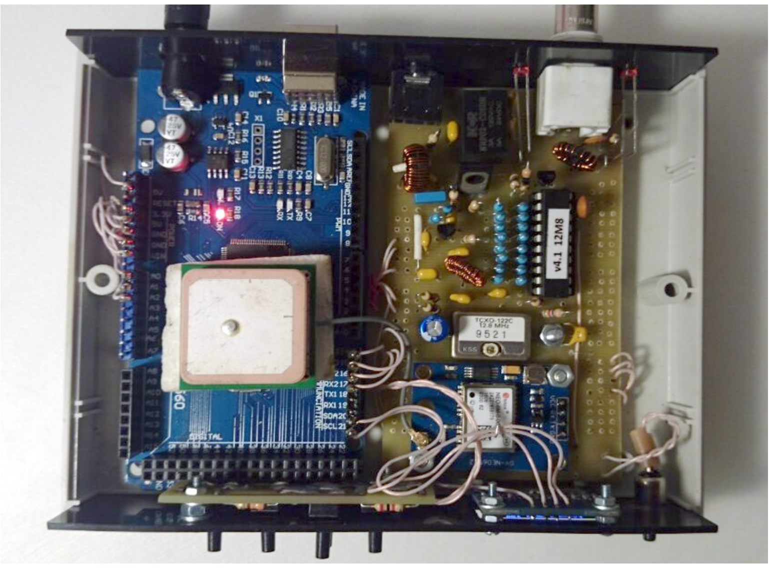
The Arduino board needs to be connected:

- to DDS-synthesizer RN3AUS
- to the GPS module
- to the LCD module (installed as a "mezzanine" over the Arduino board) or to the OLED display and keyboard.
- To RTC module (if available)

The connection method depends on which case you plan to use. The RN3AUS DDS synthesizer board is designed in the "shield" form factor for the Arduino Mega 2560, and the GPS module is installed on the same board. Thus, in the simplest case, the assembled kit will consist of three boards. Lower - Arduino Mega. A DDS synthesizer board is installed on top of it. Finally, the top board is an LCD shield or a DIY board with an OLED display and keyboard.



It is possible to arrange the boards side by side by connecting flexible connectors.



Device management. Menu system.

After turning on the device, the software is loaded, during which it searches for connected devices - a synthesizer, LCD or OLED display, RTC-module.

Upon completion of the launch, the device waits for the exact time to be set: manually or from the GPS module.

The operation and settings of the device are controlled using a simple menu system consisting of two levels: the current (operational) settings required for direct operation, and the permanent settings that do not need to be changed frequently.

Moving through the menu items is carried out by arrows "LEFT" and "RIGHT". Change of values by arrows "UP" and "DOWN". Entering a value, starting and stopping transmission, etc. - with the "SELECT" button.

The first level menu structure (operational):

SET TIME	BEACON	SETUP…	CW	OP-32	WSPR-2	WSPR-15	JT-9	FST4	QRSS	DFCW
Set exact time UP / DW - enter minutes, SELECT - set to 00 sec	Configure / start the cyclogram of the beacon	SELECT - enter the settings menu	UP / DW - frequency for CW, transmit by telegraph key.	UP / DW - frequency, SELECT - start / stop TX	UP / DW — frequency change, SELECT - start / stop TX The transmission will start at the correct time.				UP / DW - frequency change, SELECT - start / stop TX	UP / DW - frequency change, SELECT - start / stop TX

HELL	MSG QRSS	MSG DFCW	MSG HELL	MSG JT-9	MSG FST4	DOT QRSS	DOT DFCW	DOT HELL	DOT JT-9	DOT FST4
UP / DW - frequency change, SELECT - start / stop TX	UP / DW - entering characters with Morse code (UP-dash, DW-dot, on the indicator it is displayed in DFCW-style) SELECT - enter / delete the typed character				UP / DW selection from the list SELECT record	UP / DW - select the dot duration from the list, SELECT - remember this				

Структура меню второго уровня (настроек):

Exit…	CALL	QTH	PWR	FREQ GEN	LF/LF2	FREQ in LF/FREE	TIMMER CORRECTIONS	GPS	USB-GPS LINK
Exit the settings menu, return to the main menu	Callsign	QTH input	Power input UP / DW - selection from the list, SELECT - remember this	Selecting the frequency of the DDS reference generator from the list, UP / DW. SELECT-remember	On / off doubling the output frequency UP / DW, SELECT - remember	On / off prohibition on going beyond 135.7-137.8 kHz UP / DW, SELECT - remember	Entering the correction value for the Arduino system clock. UP / DW - changes, SELECT - remember	Viewing GPS Messages	Connecting a computer to the GPS module
	UP / DW - entering characters with Morse code (UP-dash, DW-dot, on the indicator it is displayed in DFCW-style) SELECT - enter / delete the typed character								

USB-SYNT LINK	LOAD FST4 TONES	CLEAR EEPROM
Connecting a computer to DDS	Downloading tones from computer for FST4 / 4W. At the end of the download SELECT - write tones to EEPROM	Clear EEPROM (factory reset) UP / DW - Yes / No selection, SELECT - execute

Entering messages

Messages are entered by typing each of the characters with Morse code. In this case, the "UP" button corresponds to a dash, "DOWN" corresponds to a dot. The display of typed characters on the screen is done in the "DFCW style".

For example, to enter the number 7, type: UP, UP, DW, DW, DW and press SELECT. The screen will display: **- - _ _ _**

Allowed character table:

0	- - - - -	C	- _ - _	O	- - -
1	_ - - - -	D	- _ _	P	_ - - _
2	_ _ - - -	E	_	Q	- - _ -
3	_ _ _ - -	F	_ _ - _	R	_ - _
4	_ _ _ _ -	G	- - _	S	_ _ _
5	_ _ _ _ _	H	_ _ _ _	T	-
6	- _ _ _ _	I	_ _	U	_ _ -
7	- - _ _ _	J	_ - - -	V	_ _ _-
8	- - - _ _	K	- _ -	W	_ - -
9	- - - - _	L	_ - _ _	X	- _ _ -
A	_ -	M	- -	Y	- _ --
B	- _ _ _	N	- _	Z	- - _ _
SPACE	- _ _ _ -	/	- _ _ - _		

You can erase the mistakenly entered character by pressing SELECT again.

Frequency setting.

Frequency setting is done individually for each operating mode. You can change the frequency using the UP - increase and DW - decrease buttons. At the first pressings, the Hertz fractions are changed, at subsequent (or holding the button pressed) the frequency step gradually increases - units of Hz, tens of Hz, hundreds of Hz. When inverting the direction of frequency change, the step is reset to the minimum. This allows you to set the required frequency as accurately as possible with a minimum number of button presses.

If the FREQ in LF mode is enabled (in the SETUP ... submenu), then for each mode the corresponding frequency limits will be in effect, upon reaching which the frequency stops changing. If FREQ is FREE is selected, then there are no such limits and you can set any desired frequency.

When the SELECT button is pressed, the set frequency is memorized in the EEPROM and the transmission of the selected mode on this frequency begins, which can be stopped by pressing SELECT again.

Setting the exact time.

Many digital modes (WSPR, JT-9, FST4) require accurate timing to operate. Therefore, after switching on, the system expects to enter the exact time either manually or from the GPS module. The time is displayed on the screen in the format MM: SS, that is, minutes and seconds of the current hour. This is done for reasons that for all digital modes it is important to link to the exact time within the current hour, that is, to minutes and seconds.

The exact time is entered manually as follows: using the UP / DW buttons set the closest minute value (for example, 18), while the seconds field remains equal to: 00. Then they wait for the time on the reference clock to show HH: 18: 00 and press SELECT. The watch starts to run.

If the device has an RTC module, then the current time value will be read from it, you just have to make sure that the time is displayed correctly. If entered manually, the time stored in the RTC will also be corrected.

If a GPS module is connected, a blinking "GPS" icon will be displayed on the screen. When the module catches satellites and receives the exact time, the flashing inscription “GPS =” will be displayed on the screen and the device time will be set automatically.

Correction of the system timer.

Depending on the specific model of the Arduino Mega 2560 board you purchased, the accuracy of its system timer may depend. Often clocking is carried out from a simple crystal, in other cases a TCXO generator is installed on the board. Of course, preference should be given to better quality boards. However, the program provides the ability to calibrate the system timer. This is necessary for the correct formation of the durations of the messages in each of the transmission modes, which can quite sensitively affect the quality of reception in all digital modes.

Initially, you need to measure the amount of "drift" of the system timer. This will require an accurate watch ("stopwatch" in your smartphone). In addition, you need to ensure that the GPS module does not receive a signal (cover the GPS antenna with a piece of foil). They turn on the device, manually set the exact time and mark exactly ten minutes on the clock on the device screen. The stopwatch is used to determine how much time has actually passed.

Next, find the difference **deltaT = Ttrue - Tarduino**.

If the Arduino clock runs exactly, then deltaT = 0. If the Arduino clock is behind, then deltaT> 0; otherwise when they are in a hurry deltaT <0.

A more accurate result can be obtained if the measurement period is increased to 30 minutes or an hour. The found value must be recalculated to a 10 minute interval. For example, in 30 minutes the time difference was 12.3 seconds, then deltaT = 12.3 * 10/30 = 4.1 seconds. It is this value that needs to be entered.

Another way to measure deltaT without disconnecting the GPS module is to transmit to the OP-32. The transmission time should be exactly 1957.8 seconds. If it turned out, for example, 1962 seconds, then

deltaT= (1957,8 - Tarduino) * 600 / 1957,8 =

= **(1957,8 – Tarduino) * 0.306**

In our example, we get (1957.8-1962)*0.306=-1,258 сек

Now the found calibration value must be entered in the settings menu: Setup ... - Timer Correction, use the UP or DW arrows to set the found correction value, press SELECT. Now it is useful to restart the program and measure the accuracy of the system timer again. If everything was done correctly, the difference between the clock rates will be vanishingly small. If, on the contrary, the difference has increased, it means that you made a mistake when entering with the correction sign.

The correction procedure usually needs to be performed only once, when preparing the device for operation.

Preparation of the cyclogram of the beacon.

The device can work as a stand-alone multi-mode beacon. To prepare the cyclogram, the BEACON menu item is used. Use UP / DW arrows to select the required mode or command. The SELECT button confirms the mode selection or executes the command. The entered commands and modes are displayed in the second line as a short record. The maximum length of this record is 15.

Modes / Commands:

load		Command, load sequence diagram from EEPROM
run		Command, cyclogram start (beacon launch)
QRSS	**Q**	mode QRSS
DFCW	**D**	mode DFCW
HELL	**H**	mode HELL
OP32	**O**	mode OP32
WSPR	**w**	mode WSPR-2
WR15	**W**	mode WSPR-15
JT-9	**J**	mode JT-9
FST4	**F**	mode FST4/FST4W
RPT	**R**	Command to repeat the sequence diagram from the beginning
save		Command, write the sequence diagram to the EEPROM memory
clr		Command, erase the last mode
T=00	**T**	Standby hour start 00:00
T=30	**t**	Half hour standby time 00:00 or 00:30

For example, we can type the following sequence:

T O w w W R

After starting the beacon (with the run command), the following will happen:

- waiting for the start of the hour 00:00
- OP32 transmission on the frequency set for this mode. (End of show at 32:38)
- WSPR-2 transmission with waiting for the start of the next even minute (34:00)
- WSPR-2 transmission with waiting for the start of the next even minute (36:00)
- transmission of WSPR-15 with expectation of the beginning of a minute in multiples of 15 (00:00 or 15:00 or 30:00 or 45:00). In this case, 45:00. End of transmission at 59: ss.
- return to the beginning of the sequence diagram.

The execution of the sequence diagram can be terminated at any time by pressing the SELECT button.

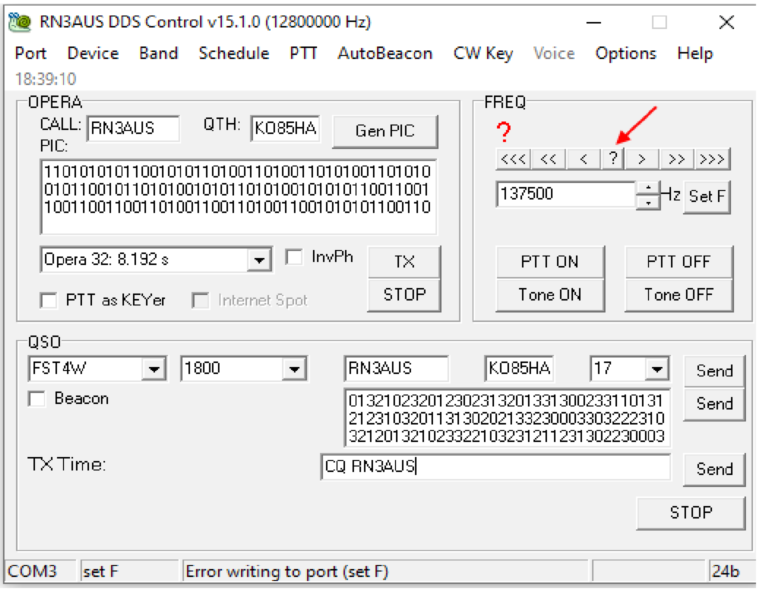
If the beacon was active and the device was suddenly turned off, then after the restart, the beacon mode is activated automatically and the execution of the sequence diagram will start from its beginning. Time correction from GPS will be carried out at the moments of waiting for the start of any of the synchronous modes. This allows the product to be operated completely autonomously, without guaranteed power supply. In any case, after the power is restored, the beacon resumes.

The sign “BCN” in the lower right corner serves as a sign of activity of the beacon.

Controlling the synthesizer from a computer.

If you connect the device to the USB port of your computer, then you can use the DDS_Ctrl program for work, which has wider possibilities compared to the program for Arduino.

In the DDS_Ctrl program, you need to select the COM port to which the connected device corresponds (usually the same port number that was determined in the Arduino development environment when we “uploaded” the sketch). Further, in the frequency setting field, you can press the "**?**" Button, or do the same through the Help menu - DDS version.



RN3AUS DDS Control v15.1.0 (12800000 Hz)

18:39:10

OPERA

CALL: RN3AUS QTH: KO85HA

PIC:

```
110101010110010101101001101001101010011010101
0101100101101010010101101010010101011001100 1
1001100110011010011001101001100101010110011 0
```

Opera 32: 8.192 s

□ InvPh

□ PTT as KEYer

□ Internet Spot

FREQ

137500 Hz

QSO

FST4W 1800

□ Beacon

RN3AUS KO85HA 17

```
0132102320123023132013313002331101 31
2123103201131302021332300033032223 10
3212013210233221032312112313022300 03
```

TX Time:

CQ RN3AUS

COM3 | set F | Error writing to port (set F) | 24b

The message «DDS not connected» may initially appear. Repeat the command 1-2 more times. The message “DDS version 4” appears and the display shows the currently set synthesizer frequency. Check that the synthesizer is controlled by sending the PPT ON - PTT OFF commands, you should hear the synthesizer's PTT relay actuation.

The device will display “USB link to SINTH”. The synthesizer is now under direct control from the computer.

At the end of the use of external control, you can either restart the device, or exit the specified mode with the LEFT / RIGHT buttons and then exit the settings menu (Exit ...).

Manual input of FST4 / FST4W tones.

To work in the FST4 / FST4W mode, it is necessary to enter the previously prepared tones into the device memory. Unfortunately, the process of encoding a message in these modes is rather complicated and there is not enough Arduino memory to implement it. Therefore, the calculation of the sequence of tones for the desired message must first be performed on the computer. This can be done using the DDS_Ctrl.exe program, or using the

fst4sim.exe utility that was included in the WSJT-X2.3 distribution package. (The same utility is included in the archive with the DDS_Ctrl program in the fst4 folder, along with the library files: libfftw3f-3.dll, libgcc_s_dw2-1.dll, libgfortran-4.dll, libquadmath-0.dll and libwinpthread-1.dll).

Create a batch file fst4w_make_tones.bat containing the line:

```
fst4sim "RN3AUS KO85 20" 60 1500 0.0 1 0.1 1.0 0 0 T > fst4tones.txt
```

(You must enter your details in quotation marks - callsign, locator and power level - as for WSPR). This format corresponds to the FST4W (beacon) mode.

After starting the batch file, the fst4tones.txt file is formed:

```
Message: RN3AUS KO85 20                    W: T iwspr: 1
f0: 1500.000   DT:  0.00   hmod:     1   TxT:  51.8   SNR:   0.0

Message bits:
110001111000111001101100011010011000001110100110000110011101110010001110000000
000000000000000000000000

Channel symbols:
0132102320
1230231320
1331300233
1301312123
1032012030
2302111233
3131330030
0011310132
1023131132
3001000103
0223130031
0033231032
0131332103
1022130330
3122111023
1001321023
```

We need Channel symbols - this is the desired sequence of 160 tones.

To send arbitrary messages in FST4 mode (for QSO), you need to correct the line, for example:

```
fst4sim "CQ RN3AUS KO85" 60 1500 0.0 1 0.1 1.0 0 0 F > fst4tones.txt
```

Tones for this message will be generated.

Manual input of tones is provided in the **FST4 TONES** menu. The input principle is similar to that described when creating the cyclogram of the beacon. Use the **UP / DW** arrows to select an item from the list, use the **SELECT** button to execute it. Tones are designated by numbers 0 ... 3. Having selected the next tone, press SELECT. The second line will show the tones entered and their number. If a tone is entered by mistake, it can be erased with the **<-del** command. At the end of dialing (or in its process, so as not to lose the entered data), the sequence of tones must be written to the EEPROM by selecting and executing the **SAVE** command. You can load the previously typed tones to continue editing with the **LOAD** command. You can erase the typed line with the **ERASE** command (Attention! The entire line will be erased entirely! Do not lose your work!). However, only the contents of the line in RAM are erased, the information in the EEPROM does not change. If the command was given in error, you can reload the previously saved line with the **LOAD** command.

Loading tones FST4 / FST4W.

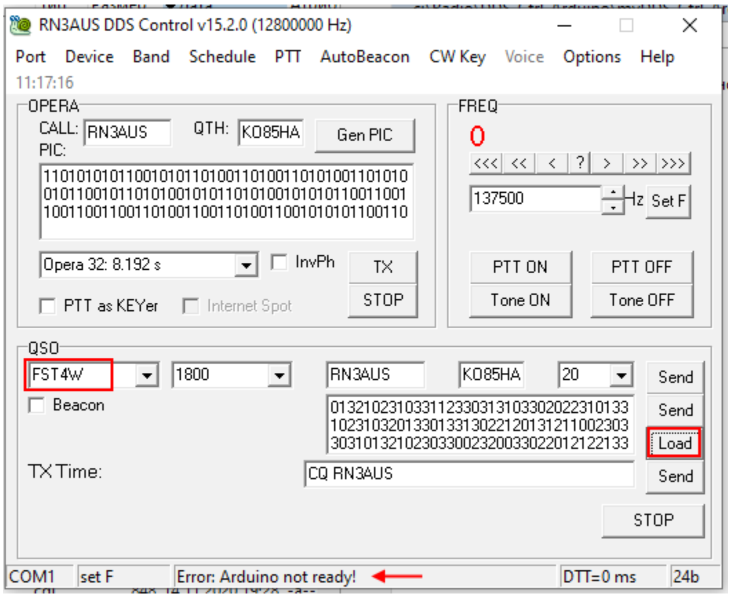
It takes a long time to enter FST4 tones manually, you need to enter 160 characters without errors. You can speed up the process by downloading these tones from your computer. To do this, enter the SETUP ... - LOAD FST4 TONES menu.

Having prepared the tones on the computer, you need to send them to the appropriate COM port in the form of a line starting with the character Z:

Z0321020302032231231233012.....

The Z flag must be followed by 160 digits representing the FST4 / 4W tones. In response, Arduino returns the received characters to the computer, starting with the Z flag. Any characters other than 0 ... 3 are ignored and are not stored in memory.

DDS_Ctrl.exe supports FST4 tone recording mode in Arduino. To do this, use the Load button next to the field containing the generated FST4 tones. The status bar of the program displays the progress of the exchange with Arduino.



RN3AUS DDS Control v15.2.0 (12800000 Hz)

11:17:16

CALL: RN3AUS QTH: KO85HA

PIC:

```
1101010101100101011010011010011010100110101 0
0101100101101010010101101010010101011001100 1
1001100110011010011001101001100101010110011 0
```

Opera 32: 8.192 s ☐ InvPh

☐ PTT as KEYer ☐ Internet Spot

0

137500 Hz

FST4W 1800 RN3AUS KO85HA 20

☐ Beacon

```
01321023103311233031310330202231013 3
10231032013301331302212013121100230 3
30310132102303300232003302201212213 3
```

TX Time: CQ RN3AUS

COM1 | set F | Error: Arduino not ready! ← | DTT=0 ms | 24b

As characters are received, the display will indicate an increasing counter and the contents of the tone bar. When the line is received, the counter will show 160. To write tones to the EEPROM, press SELECT. Otherwise, the tones can be entered again by sending a new line from the computer with the Z flag.

Once the tones have been entered and written to the EEPROM, exit the setup menu.

Work in CW mode.

Manual mode.

The device allows you to work with a classic telegraph. The XS1 connector is used to connect a vertical telegraph key.

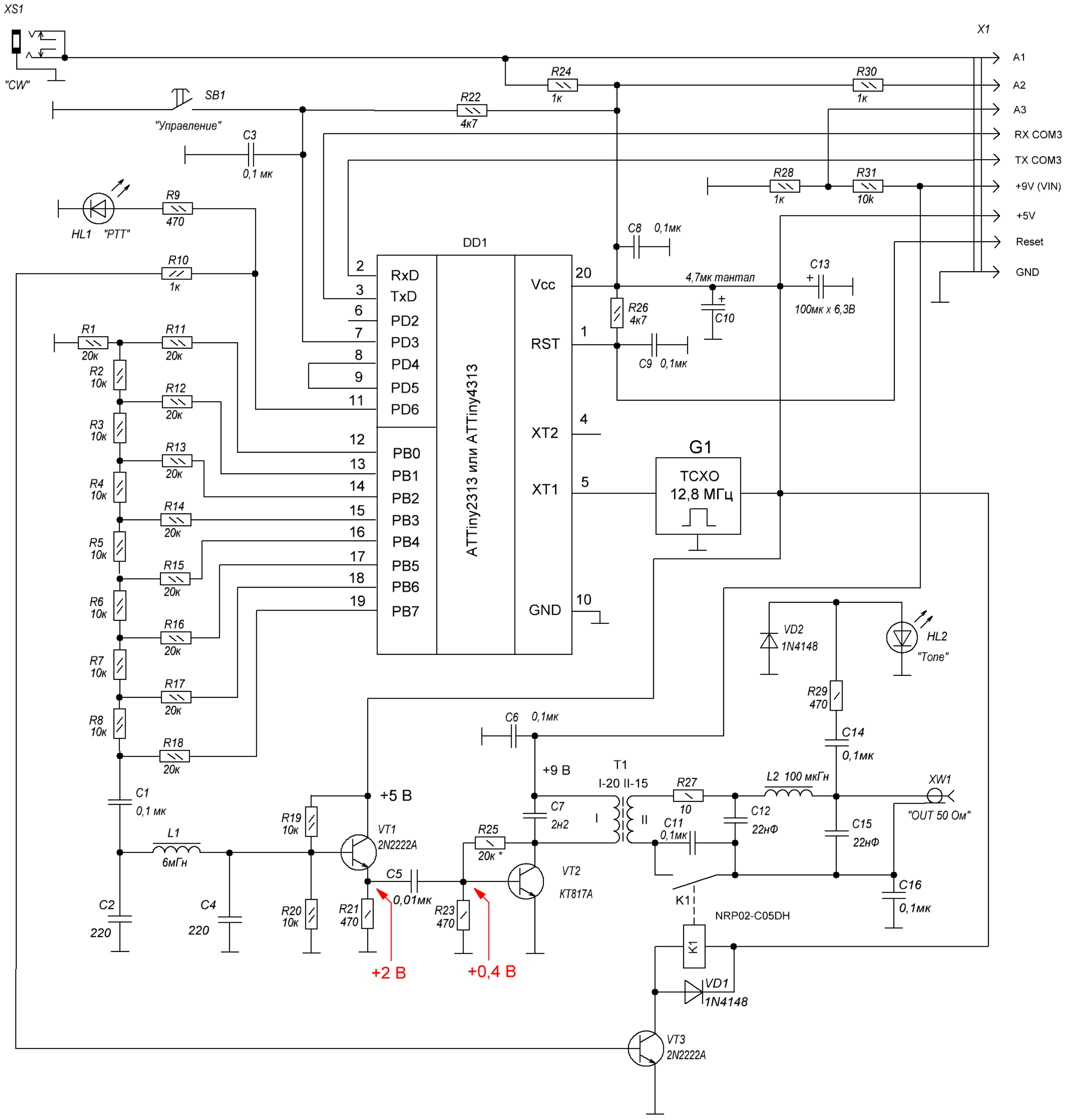
In the main menu, go to CW mode. Use the UP / DW buttons to set the desired transmission frequency, press SELECT.

Then you can start manipulating the telegraph key, while PTT will be automatically turned on and then turned off if there is no manipulation for more than 0.5 seconds (QSK).

Auto mode.

To transmit a CW beacon or short identification marker between digital mode transmission cycles, select a dot duration of 0 sec in the QRRS DOT menu, and the message will be transmitted at a speed of 10 wpm. The message text is entered in the MSG QRSS menu and has a maximum length of 15 characters.

DDS RN3AUS synthesizer circuit



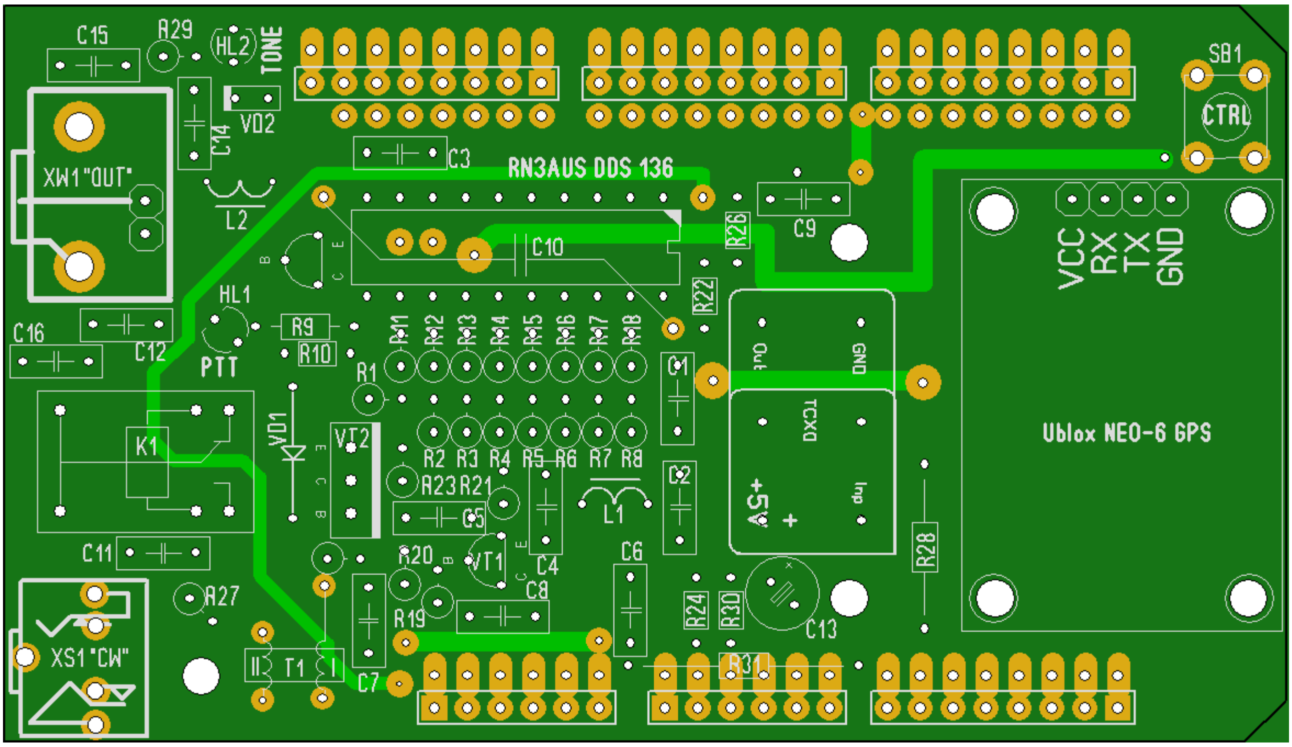
Transformer T1 on the K10x6x4.5 2000NM ring: I - 20 turns, II - 15 turns.

Inductance L1 (6 mH) on the K10x6x3 2000NM ring: 120 turns.

List of synthesizer elements.

C5	= 1 x 0,01uF
C7	= 1 x 2,2nF
C10	= 1 x 4,7uF tantal
C13	= 1 x 100uF x 6,3B
C12,C15	= 2 x 22nF
C2,C4	= 2 x 220
C1,C3,C6,C8,C9,C11, C14,C16	= 8 x 0,1uF
DD1	= 1 x ATTiny 2313 or ATTiny 4313
G1	= 1 x 12,8 MHz TCXO
HL1	= 1 x "PTT"
HL2	= 1 x "Tone"
K1	= 1 x NRP02-C05DH
L1	= 1 x 6mH (120 turns on the ring K10X6X3 2000HM)
L2	= 1 x 100 uH (17 turns on the ring K10X6X3 2000HM)
R27	= 1 x 10
R25	= 1 x 20k*
R1,R11,R12,R13, R14,R15,R16, R17,R18	= 9 x 20k
R10,R24,R28, R30	= 4 x 1k
R2,R3,R4,R5,R6, R7,R8,R19,R20, R31	= 10 x 10k
R22,R26	= 2 x 4k7
R9,R21,R23,R29	= 4 x 470
SB1	= 1 x "Control"
T1	= 1 x K10x6x4.5
VD1,VD2	= 2 x 1N4148
VT2	= 1 x KT817A (or BD149)
VT1,VT3	= 2 x 2N2222A
XS1	= 1 x Stereo ST-111 3,5 mm ("CW")
XW1	= 1 x BNC-7043 ("OUT 50 Ohm")

Arrangement of elements on the synthesizer board.



Notes:

Resistor R25 is next to VT2 (below it), not signed.

Capacitor C10 is installed on the back of the board.

When installing transistors VT1 and VT3, be guided by the designation of electrodes C, B, E! The pinout on the board (case orientation) on the board corresponds to the BC547 transistor. For 2N2222 transistors, depending on the manufacturer, the pinout may differ! Most often there are 2N2222 with reverse pinout (E-B-C), while on the board they are placed "back to front" relative to the orientation of the transistor case indicated on the board. Be careful!

Under the microcontroller socket there are contact pads connected to pins 8 and 9 of the MK. Install a jumper between them on the back of the board. When the jumper is installed, the synthesizer works normally.

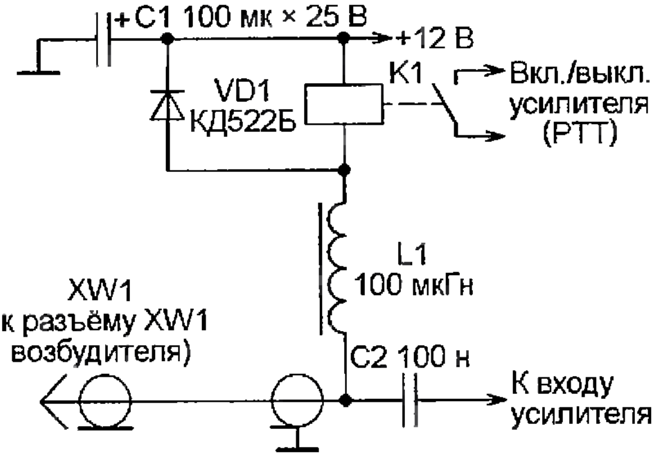
If the jumper is removed, the PTT auto-off mode will be disabled if there are no commands for more than 20 minutes (protection against hangup of the control program). In this case, however, it becomes possible to transmit a long (many hours) coherent carrier.

Establishing a synthesizer is usually reduced to the selection of such a value for the resistor R25 so that the voltage based on VT2 in the absence of a signal is 0.3-0.4 V (when the power is + 9V), the transistor is closed.

The primary winding of the transformer T1 (20 turns) has an inductance of about 600 μH and is tuned into resonance by a capacitor C7 at a frequency of 137 kHz. In this case, on the VT2 collector, the signal should be in the form of sinusoidal impulses with a cutoff, without overshoot. If you get a different value of the inductance of the primary winding, select C7.

Please note that the secondary winding of T1 (15 turns) and the body of the coaxial connector XW1 are galvanically isolated from the common bus of the product and are grounded at HF through C16. This decoupling is made to reduce possible interference when operating the synthesizer with a power amplifier.

The control of turning on the power amplifier (PTT) is carried out using the relay K1, which, with its contacts, closes the DC circuit, bypassing the capacitor C11. In this case, the PTT signal is fed over the same coaxial cable as the 137 kHz signal. At the input of the transmitter, there must be a circuit for separating the RF signal and PTT:



As can be seen from the diagram, the transmitter also contains a relay, the winding of which is powered by + 12V, and its second end is switched to ground through the decoupling choke, the central core of the coaxial cable and then relay contacts K1 in the synthesizer.

This application of a PTT signal to the transmitter seems a little complicated, but there is an explanation for this. Most often, it is necessary, together with the synthesizer, to be located at a distance from a powerful transmitter in order to avoid interference and prolonged presence of the operator in an HF field of high intensity (if the extension coil is located at the transmitter). In this situation, reducing the number of wires connecting the transmitter and synthesizer to one is very convenient.

Coil L1 (120 turns) with capacitors C2, C4 form a low-pass filter with a cutoff frequency of about 200 kHz. The quality of this low-pass filter (Q-factor of the L1 coil) determines the degree of presence of "jitter", phase jitter, of the generated signal. You can use a ready-made coil with the specified inductance. If, when adjusting, the signal level at the output of the low-pass filter turns out to be small (fractions of a volt), then, obviously, the inductance L1 turned out to be too large.

The output of the synthesizer amplifier is also equipped with a low-impedance LPF L2, C12, C15. To ensure the stability of the amplifier when operating on an unmatched load (idle or short circuit), a resistor R27 is installed in front of the low-pass filter. It is permissible to use a ready-made choke of 100 µH. If the signal at the XW1 connector is small (fractions of a volt) and the HL2 LED does not light up, while there are pulses with an amplitude of several volts on the VT2 collector, this is a sign of too large inductance L2. The cutoff frequency of the LPF turned out to be lower than the required one and the 137 kHz signal is suppressed.

You can check the assembled synthesizer offline, without connecting to the Arduino. To do this, the + 5V and + 9V contacts of the board are supplied with + 5V voltage (the output stage can also operate at a reduced voltage). Then, by short pressing the SB1 button, the generation of the carrier 137500 Hz is turned on, by another short press, the carrier is turned off. Long press (more than 2-3 seconds) turns on / off PTT, relay K1 is activated.

If there is an uncertain operation of the SB1 button, the capacitance of the capacitor C3 should be increased - perhaps there is contact bounce. It is also important to have a tantalum capacitor C10, installed on the back of the board as close as possible to the power and ground pins of the microcontroller - parasitic RF pulses may be present on the power bus that disrupt the normal operation of the MC.

Connection diagram for installation with flexible connectors.

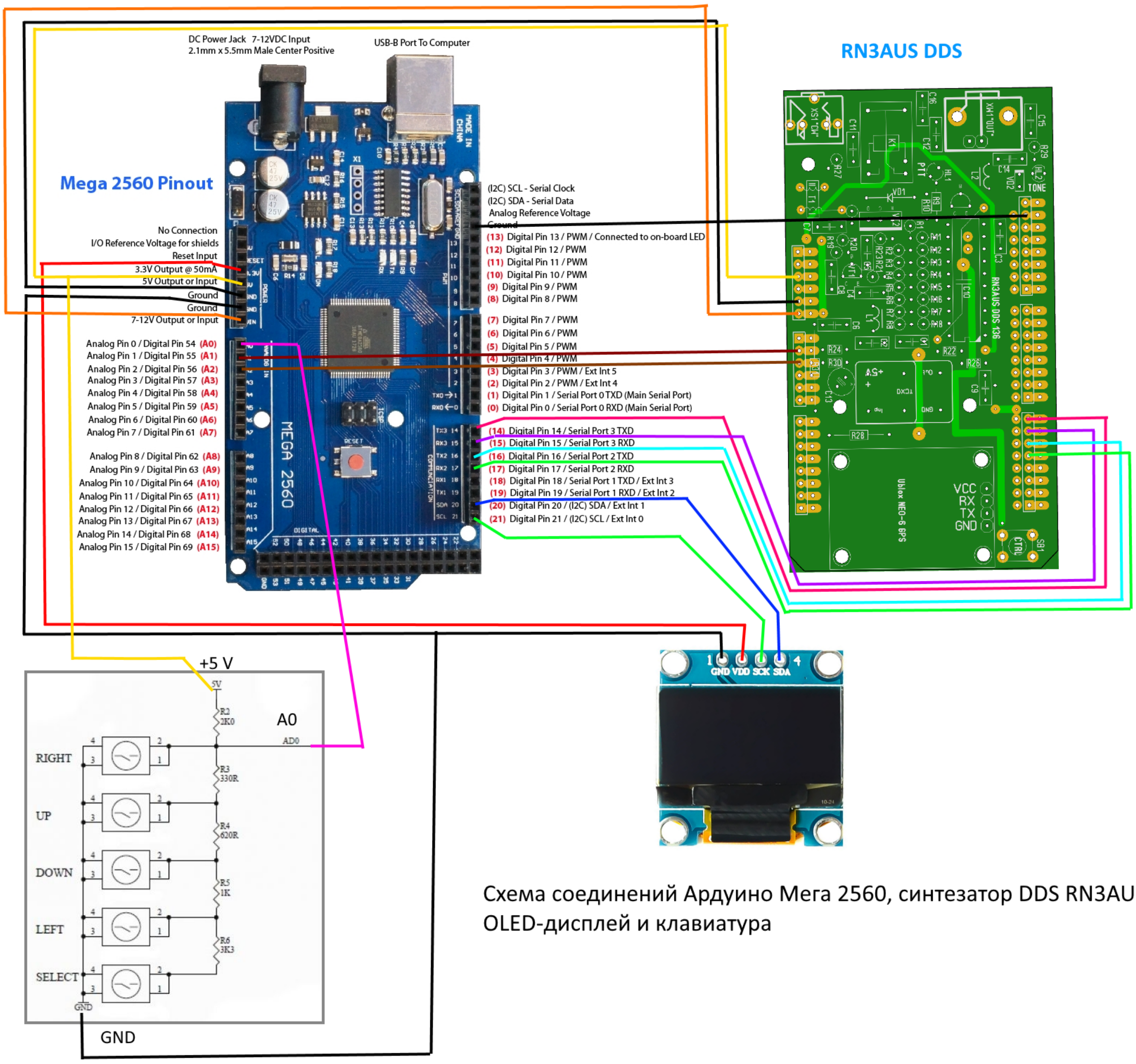
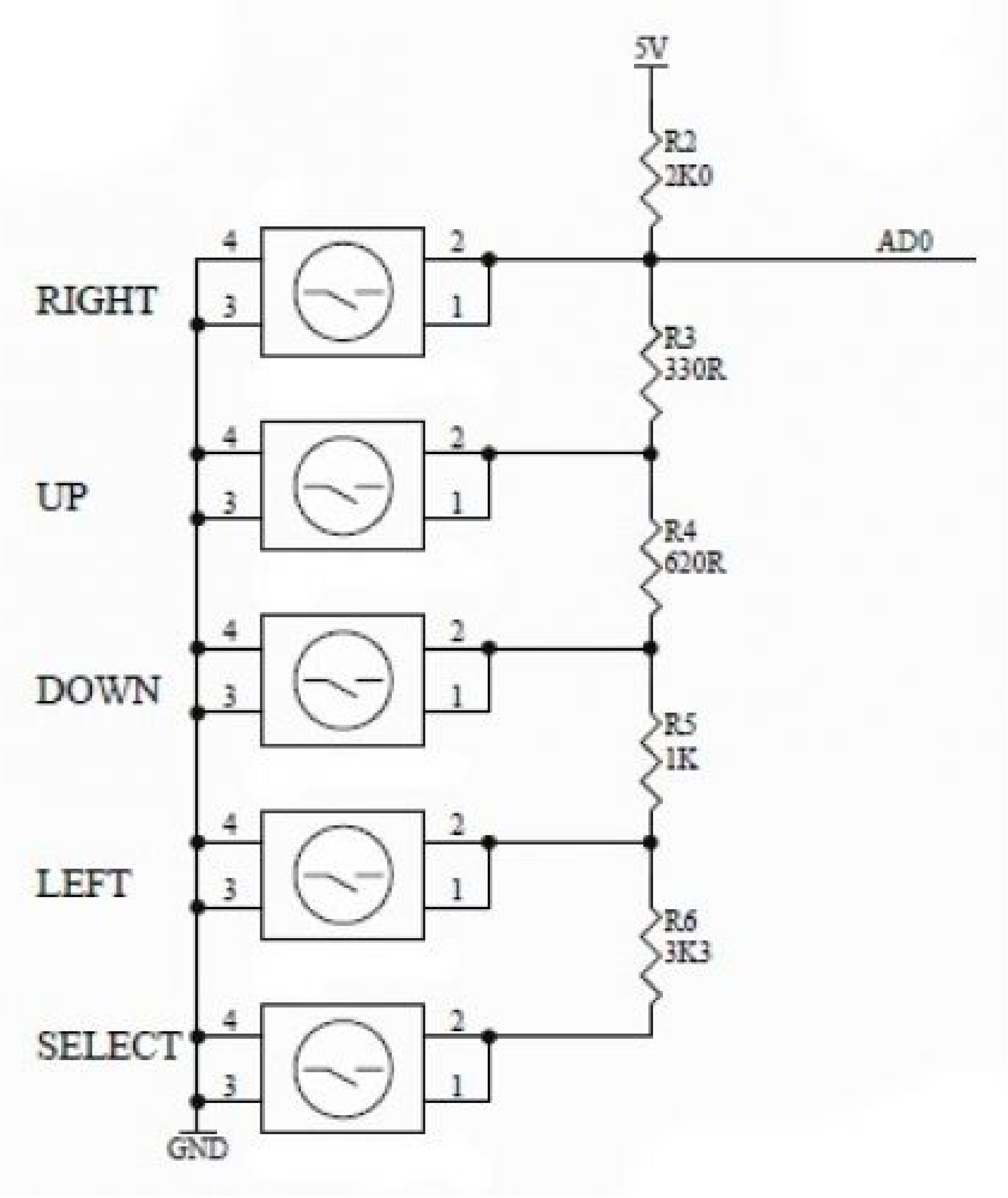

RN3AUS DDS
Mega 2560 Pinout
DC Power Jack 7-12VDC Input
2.1mm x 5.5mm Male Center Positive
USB-B Port To Computer
No Connection
I/O Reference Voltage for shields
Reset Input
3.3V Output @ 50mA
5V Output or Input
Ground
Ground
7-12V Output or Input
Analog Pin 0 / Digital Pin 54 (A0)
Analog Pin 1 / Digital Pin 55 (A1)
Analog Pin 2 / Digital Pin 56 (A2)
Analog Pin 3 / Digital Pin 57 (A3)
Analog Pin 4 / Digital Pin 58 (A4)
Analog Pin 5 / Digital Pin 59 (A5)
Analog Pin 6 / Digital Pin 60 (A6)
Analog Pin 7 / Digital Pin 61 (A7)
Analog Pin 8 / Digital Pin 62 (A8)
Analog Pin 9 / Digital Pin 63 (A9)
Analog Pin 10 / Digital Pin 64 (A10)
Analog Pin 11 / Digital Pin 65 (A11)
Analog Pin 12 / Digital Pin 66 (A12)
Analog Pin 13 / Digital Pin 67 (A13)
Analog Pin 14 / Digital Pin 68 (A14)
Analog Pin 15 / Digital Pin 69 (A15)
(I2C) SCL - Serial Clock
(I2C) SDA - Serial Data
Analog Reference Voltage
Ground
(13) Digital Pin 13 / PWM / Connected to on-board LED
(12) Digital Pin 12 / PWM
(11) Digital Pin 11 / PWM
(10) Digital Pin 10 / PWM
(9) Digital Pin 9 / PWM
(8) Digital Pin 8 / PWM
(7) Digital Pin 7 / PWM
(6) Digital Pin 6 / PWM
(5) Digital Pin 5 / PWM
(4) Digital Pin 4 / PWM
(3) Digital Pin 3 / PWM / Ext Int 5
(2) Digital Pin 2 / PWM / Ext Int 4
(1) Digital Pin 1 / Serial Port 0 TXD (Main Serial Port)
(0) Digital Pin 0 / Serial Port 0 RXD (Main Serial Port)
(14) Digital Pin 14 / Serial Port 3 TXD
(15) Digital Pin 15 / Serial Port 3 RXD
(16) Digital Pin 16 / Serial Port 2 TXD
(17) Digital Pin 17 / Serial Port 2 RXD
(18) Digital Pin 18 / Serial Port 1 TXD / Ext Int 3
(19) Digital Pin 19 / Serial Port 1 RXD / Ext Int 2
(20) Digital Pin 20 / (I2C) SDA / Ext Int 1
(21) Digital Pin 21 / (I2C) SCL / Ext Int 0
MEGA 2560
Ublox NEO-6 GPS
VCC
RX
TX
GND
1 GND VDD SCK SDA 4
+5 V
5V
R2 2K0
A0
AD0
R3 330R
R4 620R
R5 1K
R6 3K3
RIGHT
UP
DOWN
LEFT
SELECT
GND
GND

Схема соединений Ардуино Мега 2560, синтезатор DDS RN3AUS, OLED-дисплей и клавиатура

Note: The Ublox NEO-6M GPS module is installed on the synthesizer board.

Keyboard layout.


5V
R2
2K0
AD0
RIGHT
R3
330R
UP
R4
620R
DOWN
R5
1K
LEFT
R6
3K3
SELECT
GND
4
3
2
1