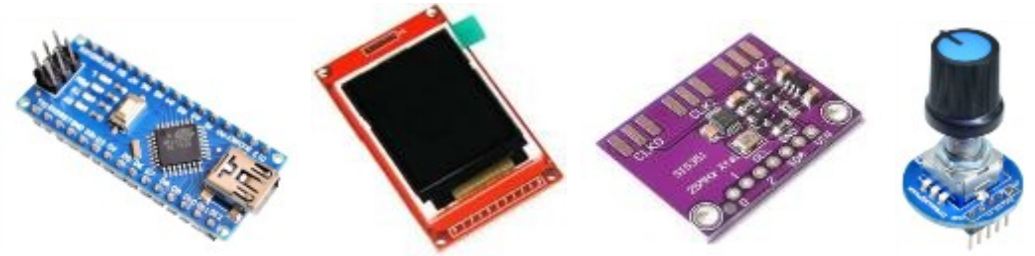
GEBETRACK

Talking to friends, I noticed that some of them were in need of an RF generating tool, using the SI5351, to carry out checks on the possibility of installing the DDS KIT on radios other than those we have already carried out
modification by DDS.
A simple fundamental generator would be ideal, but that would be a huge waste of Arduino flash memory, so I developed a system containing some functions aimed at serving some enthusiasts of the amateur radio.
The tool was named by me with the name GEBETRACK. Junction of the words GENERATOR, BEACON AND TRACKING.
GEBETRACK can generate two independent frequencies from 1.5KHz to 280MHz in fundamental frequency or up to 999MHz using harmonics, it can also generate frequencies with 90º lag between output CLK0 and output CLK1.
In summary, the tool has:
FUNDAMENTAL generator with complete STEP between all 9 digits. (VFO without OFFSET)
OFFSET VFO generator with complete STEP between all 9 digits. (VFO with OFFSET)
Scan generator or tracking generator with width and 3 directions fully configurable on the screen.
Double BEACON generator. It can emit beacon of up to 26 letters on up to two different frequencies.
It also has a 30-minute inactivity timer to be used on the display LED to save battery when using the beacon for long periods of time using a battery.
The GEBETRACK system is complete and available for free at www.kitdds.blogspot.com
BUT I WANT TO MAKE ALL AWARE THAT THIS PROJECT IS FOR PERSONAL USE AND PRODUCTION IS PROHIBITED
COMMERCIAL, AMENDMENT OR RESALE OF THIS PROJECT.
The project is for radio amateurs enthusiasts.

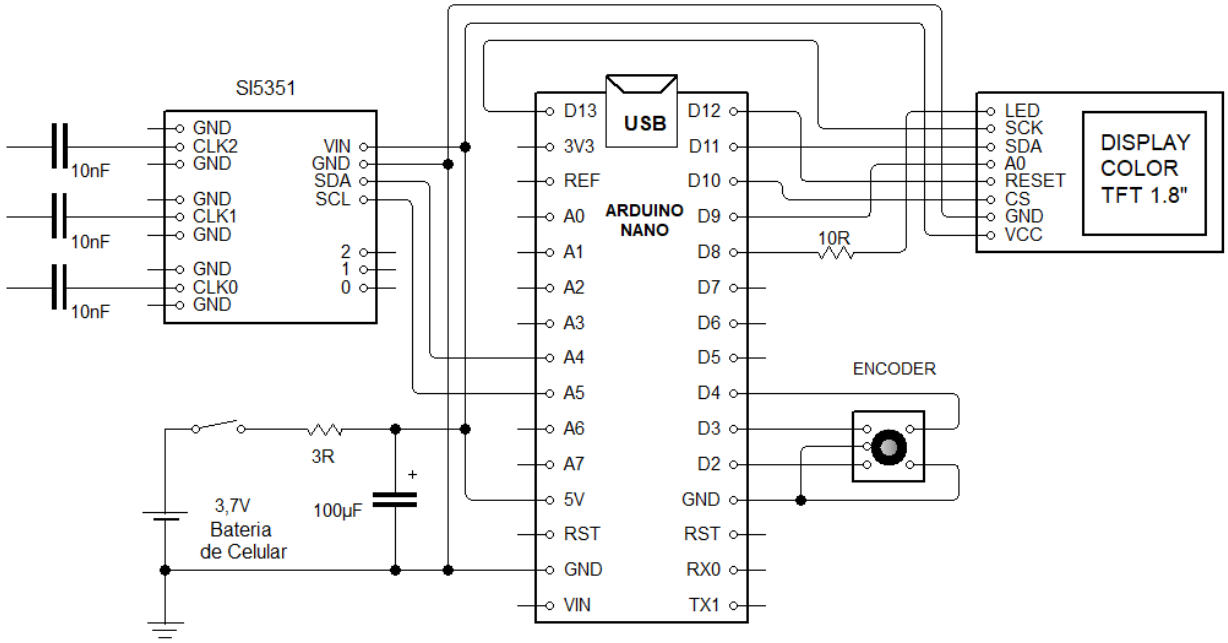
Of course, any donation for better projects will be welcome and can be made by PayPal via email: wanderlg@gmail.com

To make the project you will need:



1 - Arduino Nano
1 - 1.8 ”TFT color display
1 - SI5351 module
1 - Rotary Encoder
1 - 3.7V Lithium battery
1 - Resistor 3R - 1W (Orange, Black, Gold)
1 - 10R 1 / 4W Resistor (Brown, Black, Black)
1 - Electrolytic capacitor 100uF x 16v
Soldering iron, solder and wires.

FULL GEBETRACK CIRCUIT DIAGRAM



```
Use a capacitor from 1nF to 100nF on the 3 outputs of the SI5351, as this will
protect against accidental voltage.
```

Details of the Arduino pinouts:

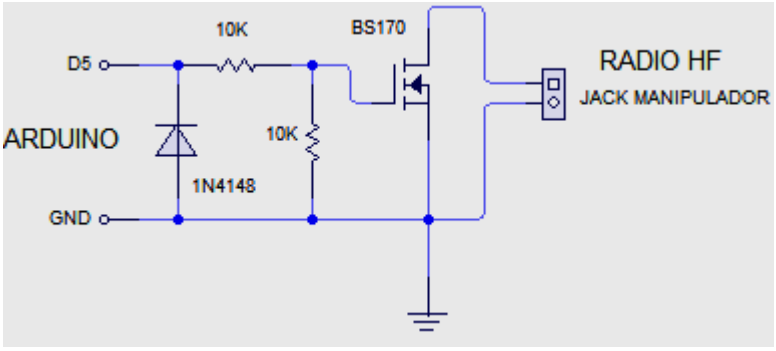
Pin D2 ---- DT Encoder
Pin D3 ---- CLK Encoder
Pin D4 ---- SW Encoder Button
Pin D5 ---- **Optional** . STATUS BEACON LED (CW oscillator output)
Pin D6 ---- **Optional** . OUTPUT CONTINUES BEACON OUTPUT (PA power supply)
Pin D7 ---- **Optional** . 800Hz TOM output
Pin D8 ---- LED Display (30 minute timer for display light)
Pin D9 ---- A0 Display
Pin 10 ----- CS Display
Pin 11 ----- SDA Display
Pin 12 ----- RST Display
Pin 13 ----- SCK Display
Pin A0 -----
Pin A1 -----
Pin A2 -----
Pin A3 -----
Pin A4 ----- SDA SI5351
Pin A5 ----- SCL SI5351
Pin A6 -----
Pin A7 -----

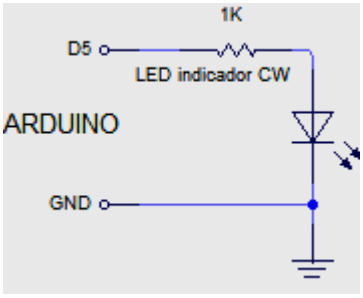
As the entire GEBETRACK circuit consumes about 50mA with LED on the display or 20mA with LED the display is off, you can use any 5V source (Ex: Cell phone charger). The lithium battery Cell phone 3.7V already contains a protection circuit that will cut the input current, as soon as the battery is fully charged. You can take any 5V charger and use it as a source for feed the entire GEBETRACK circuit, use and charge your battery safely.
Using a 3.7V Lithium cell battery, you can connect to the Arduino 5V terminal, use a resistor as suggested in the diagram, as the battery protection circuit does not control the current. Do not connect battery in the Arduino VIN. The battery, when fully charged, will have a voltage of 4.2V.
To make it easier, the battery voltage indication has 4 colors.
Blue will be displayed when the voltage is above 4.2V (this is normal, it will happen when it is with charger connected to the Arduino's USB port, as the higher voltage will stay on the Arduino and the resistor 3R will decrease the voltage on the battery. Since the system is reading the Arduino supply voltage, the voltage indication will be higher.)
Green indicates a fully charged battery.
Yellow indicates battery is discharging.
Red indicates that it is time to load the GEBETRACK. (The charge must be made with the GEBETRACK on)
NOTE: Charge the battery in this way, only if it is a cell phone and has an internal protection circuit, all good quality original batteries have this protection.

Optional Pins.

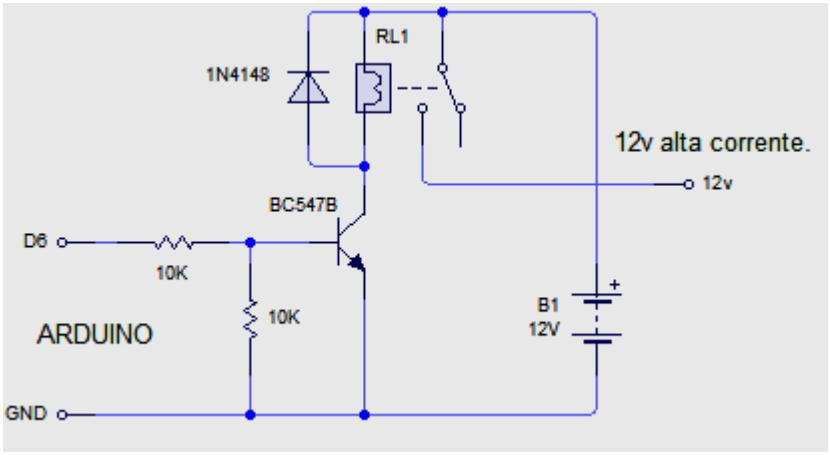
The optional pins on the Arduino 5,6,7 are completely related to the beacon. These pins are only for that enthusiast who wants to use GEBETRACK as a beacon transmitter. In this case the pins options can be used for external functions.

Pin D5 is equivalent to a woodpecker manipulating the MORSE message and can be used to arm the input of a CW handler into an HF radio. Depending on the radio model, if it is a tube, it is better to add a relay in this circuit. On pin D5, an indicator LED can be used CW to make GEBETRACK more complete.

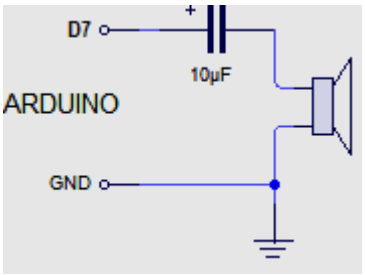




Pin D6 serves as a switch to arm an RF amplifier circuit. I will leave a circuit example with relay, but it is not necessary to relay for this type of circuit, because generally, you will not use more than 1W in a beacon, so a small mosfet will already serve as a key for a small PA circuit.



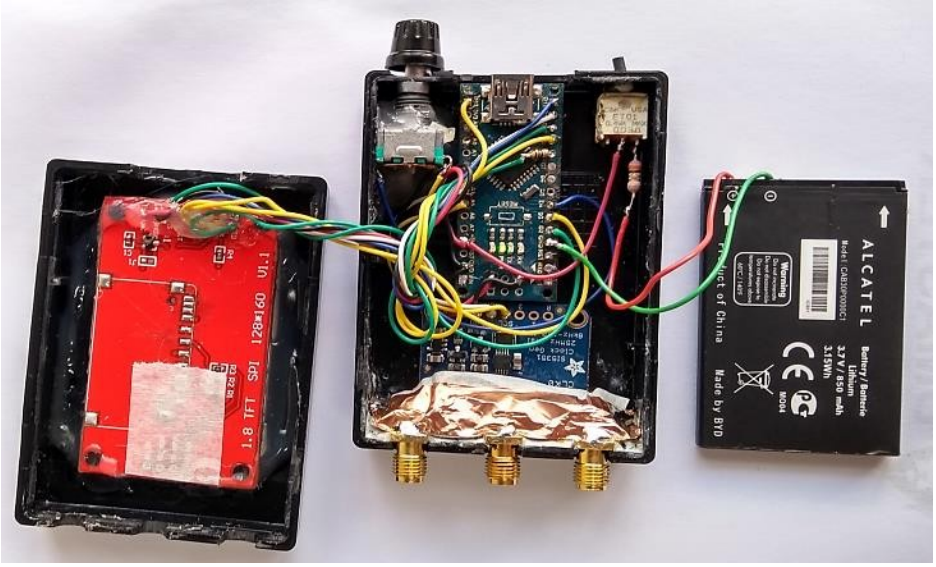
Pin D7 is only an 800Hz audible emitter of the MORSE message. A small passive buzzer.



But as already described, all of this is optional.

Mounting suggestion.

As this is a project for hobbyists, I will not go into detail on how to make it, as this is the role of the hobbysta and each one of us will have different ideas about the montage. I'll just show you a photo of how I did mine. I used the box of a small 5V power supply.



In my assembly, I did not use any of the optional pins.

Operation:

After assembly, the only mandatory configuration to be made is to adjust the reference crystal of the SI5351. But first, it will be necessary to understand how to use the encoder to browse all system resources GEBETRACK.

The circuit has a single button and that button has only two options.

Option 1 (short clicks, less than 1 second pressed).

Option 2 (long clicks, more than 2 seconds pressed).

Note that in the first photo, there is no item marked with a red stripe. When you observe any item flashing with a red stripe, it means that you are navigating between items in the menu, use the direction of the encoder switch and move between items. If you leave anything on for more than 6 seconds, the system will exit the menu automatically, as this prevents you from forgetting the system in the menu and the update of the display makes the system very slow, if you are using the mode TRACKING.

To calibrate the SI5351 reference crystal, move between the items, until the last item that is in front of the acronym CLK2, make a short click on the encoder and a white stripe will appear over the word according to the first photo. Move the encoder until the word **FUNDAMENTAL** appears , as shown in the first photo.
Now make a short click until the frequency below is white and the white stripe disappears from the word **FUNDAMENTAL** , according to the second photo. In this mode, each short click on the encoder will change the STEP steps, see the yellow color on the number, this is the number that will be moved to the
move the encoder. Now, set the frequency to 10MHZ according to the third photo. Ready.
Connect the frequency meter to the CLK2 output on the SI5351 and check if it is exactly 10MHz, if it is outside frequency, enter the menu again by holding the button for more than 2 seconds and release, roll the encoder to item **SI5351: XTAL** and short-click. You will have a screen according to the last photo. With short click you will change the steps of the STEP, change the frequency of the reference crystal until you see it on your frequency meter,
the exact 10MHz being generated at the CLK2 output of SI5351. Now to save this change and exit the menu, hold the button for more than 2 seconds and release.
Ready. You have already learned how to navigate the menu and make changes to any item you want.
To exchange the beacon message, the procedure is exactly the same.

IMPORTANT DETAILS:

The item >> **CLK0** <<, >> **CLK1** <<, **CLK0** <> **CLK1** refers to the 90 ° lag between CLK0 and CLK1.
When you choose >> **CLK0** << or >> **CLK1** << **it** means that one of the two outputs that is not described in screen, has been disabled and when you choose **CLK0** <> **CLK1** means that the system is generating the same desired frequency in both outputs at the same time, but lagging 90º.
I advise you to use the CLK2 output below, in case you do not need a 90º lag, as the CLK2 is faster and will have a better response when using the TRACKING generator.
Speaking of a TRACKING generator, it is very easy to understand and use. You will choose a frequency initial, choose the direction of the scan that can be in loops >> forward, << back, > <
back and forth, choose the final frequency, define a step for the **STEP** (in the photo below the step is in 25KHz), choose the interval between each step and turn on the TRACKING mode according to the third photo.

The scan will be generated at the CLK output that you have activated in the **TRACKING** mode **,** according to the last photo,
is being generated at CLK2. The 10MHz frequency will be in memory and will not be generated in tracking, since the scan is set to start at 20MHz and end at 30MHz, only the CLK2 output is being used to generate the scan.

You can also generate different functions at the same time.

In this example, the beacon message is being generated at the frequency of 7.021MHz using the output **CLK0** and the beacon mode in: **BEACON CW** means that the carrier at the **CLK0** output will be **oscillated** in perforated carriers in CW, only the **BEACON FM** mode , which maintains a continuous carrier, however it would be necessary to make a change in the reference crystal circuit of the SI5351, using a diode varicap, capacitors, resistors and the Arduino D7 pin. At the same time that the beacon on **CLK0** , output **CLK2** is being used as a VFO for a Cobra 148 GTL radio displaying **27,205MHz** on the display, however generating **35,005MHz** on the **CLK2** output , as it is in **VFO OFFSET** mode and the
OFFSET is set to **7,800MHz**
OFFSET can be set to positive values (adding to the VFO display frequency)
or negative values (subtracting from the VFO display frequency).
If desired, you can define the beacon on two different frequencies.
Be aware that generating the beacon at 7.021MHz and without using a low pass filter for that frequency, you will be generating in all harmonics. In other words: it is generating in
7.021MHZ 14.042MHz 21.063MHz28.084MHz
And always going up. X1, X2, X3, X4, X5, X6 ...

Almost infinite. Rsrsrsrsrs. But of course there will be no good outcome
because if you are not using an RF amplifier, the power is on average 15mW and your antenna is not is made for all these bands.

That's it.
Hug and a lot of peace.
wanderlg@gmail.com