

Enrico Miglino

Vintage Upcycling

with Raspberry Pi and Arduino



Projects based on *Element14.com* workshops
and the Vintage Tech series on *Shed Magazine*

element14

Project14

Shed

ALWAYS
DESIGNER

ELEGOO

EDICIONES BALEARES

Find full paperback on Amazon:

https://www.amazon.com/Vintage-Upcycling-Raspberry-Arduino-Makers/dp/849414121X/ref=sr_1_2?dchild=1&keywords=miglino&qid=1609785525&sr=8-2

Kindle ebook available, free on Kindle Unlimited:

https://www.amazon.com/Vintage-Upcycling-Raspberry-Pi-Arduino-ebook/dp/B08RJ85SQ1/ref=tmm_kin_swatch_0?_encoding=UTF8&qid=1609785525&sr=8-2

Vintage Upcycling

With Raspberry Pi and Arduino

Enrico Miglino

FREE PREVIEW

Copyright ©2020 Ediciones Baleares

all right reserved

ISBN-978-84-941412-1-8

“Vintage Upcycling With Raspberry Pi and Arduino”

Copyright © Ediciones Baleares 2020

Av.da Miguel Hernández 44 2/D

03700 Dénia (Alicante) Spain

e-mail edbaleares@gmail.com

All rights reserved. No part of the contents of this book may be reproduced, transmitted in any form or by any means, or translated without the written permission of the publisher.

This book is provided “as-is” and expresses the authors’ views and opinions.

The views, opinions, and information expressed in this book, including URL and other Internet website references, are subject to change without notice.

ISBN 978-84-941412-1-8

Editorial Production: Ediciones Baleares

Copyeditor: Furio Piccinini

Proofreader: Maria Antonietta Ricagno

Cover: Furio Piccinini

Photos: Enrico Miglino

Table of Content

Preface	II
Acknowledgments	III
Introduction	IV
Radio Magic	6
5.1 A First Musical Project	6
5.2 Hacking the Radio Tuner	8
The Tuner Mechanism	9
5.3 The New Components	9
5.4 The Control Board	12
5.5 The Arduino Sketch	14
LED Activity	16
The Rotary Encoder	18
The Run/Standby Pushbutton	20

Preface

A lot of fun and challenging too

The Shed magazine is predominantly a print magazine and we have been privileged to have published most of the articles here in this publication by Enrico Miglino.

These upcycling projects have been very well received by our readers who enjoy their spare time, making and creating projects for fun or folly. With electronics now playing such a huge, huge part in our everyday lives, so many more of our readers are enjoying projects of this type where they can test their skills and expand their knowledge horizons.

All these projects here do just that and we hope you enjoy them as much as our readers have. To test your skills and have fun doing so is the perfect way to enjoy your leisure time. There is nothing like creating a project that brings on a smile and makes good use of stylish antique pieces while using contemporary, easy to access, hardware and software.

These upcycling projects are a perfect match for the electronics hobbyist and for those that appreciate that many older items still have value, use and a place with us all in the 21st century.

Enjoy.

Greg Vincent, Publishing editor
The Shed magazine

Acknowledgments

If you are reading this book it is also thanks to the support and sponsorship of Project14 and the element14.com community that gave me the possibility and help to create the three part workshop “Vintage Upcycling”. During the chats with Tariq while preparing the workshop script, I had the idea of writing this book.

A big thanks also to Greg and The Shed magazine where I published some of the projects covered by this work.

The presentation of professional schematics and PCB designs was made possible thanks to the kind sponsorship of Altium Designer, one of the best CAD circuit design applications.

All the 3D printed components shown in these projects were made using filament and resin 3D printers provided by Elegoo.

Introduction

by Tariq Ahmad

Community manager of Project14 at element14.com

When I think of vintage electronics, what I tend to think of the beautiful craftsmanship on my parents old Marentz receiver from the 70s accompanied with floor speakers with room filling sound. Or, maybe a vintage console television with a CRT screen that was 25 inches, which was bulky even by today's standards, even if the screen is smaller than what people are use to. Technology back then was not meant to be tucked away or hidden, it had to be made as beautiful as any piece of furniture in the room. Nowadays, the life cycle for electronics is much shorter, and the footprint it leaves is not so drastic.

When Enrico, approached me about doing a workshop around an old rotary phone, I recalled a rotary phone that I fell in love with at a neighbours house growing up. Bringing back something old and familiar, isn't just a shameless plug for nostalgia, it's a return to something beauty lost. For all its sophistication, today's technology has a short life cycle, and an almost disposable quality. This was less the case 40 or 50 years ago, has time flown by so much since then? The upcycling workshop that Enrico proposed, felt almost Proustian, in its search for time regained. Every piece of vintage technology recalled memories from era lost but not forgotten.

The idea for having workshops on the element14 community came from a desire to make electronics projects accessible to everyone. That was one of the missions of the Project14 project competitions held on the element14 community. With Enrico's support, we got together to arrange a series of workshops based on projects in this book, that involved upcycling Vintage Electronics.

What is upcycling you might ask? Upcycling, also known as creative reuse of existing material or products perceived to be of great quality or aesthetic value. For electronics hobbyists, that usually means using something like a Raspberry Pi or an Arduino, in order to make something old, new again. The idea for the Vintage Tech Workshop.

The idea for this workshop was to inspire people to take cool vintage appliances and repurpose them using a Raspberry Pi or an Arduino. If you are a fan of electronics, you likely have some older electronics lying around that are in need of a second life. One of the goals of upcycling is to prevent wasting material that could be potentially useful by making use of it. By doing so, you reduce the consumption of new raw materials in effect creating a new product; thus, reducing energy usage, air and water pollution, and even greenhouse gas emissions.

Following the 3 part workshop series on Vintage Upcycling with Raspberry Pi and Arduino that Enrico hosted, the members voted on a Recycle & Retrofit competition in the Project14 program on element14 community. Project14, is a member-driven program on the element14 community where members come up with ideas for competitions, vote on the competitions they want to participate in, and then decide the winners of those competitions. In fact, the first competition ever run on Project14 was Techno Toys, an upcycling competition, and throughout the programs lifespan, many of the most memorable electronics projects, involved upcycling.

Upcycling connects the past to present in a way that honours what we cherish from vintage electronics while improving it to make it contemporary using modern electronics such as a Pi or an Arduino. One of the goals of upcycling is have a more “creative” version of recycling, not only designing, adapting and making usable something vintage, while adding more value, and a brand-new charm. To “upcycle” an item, is to essentially update, modernise or deconstruct it. Upcycling is about aesthetics as well as substance. Upcycling can make what you once loved, contemporary while not losing any of its old world charm.

If you’re new to electronics projects, or upcycling, we hope this book inspires you to try your hand in electronics project. If you’re more experienced, we hope that the projects in this book inspire you to take on an upcycling project and make something beautiful. In this way we can make what is old, new and sustainable at the same time.

Radio Magic

5.1 A First Musical Project

This project first appeared on “The Shed” magazine issue number 92, September/October 2020 (<https://the-shed.nz/>) and has been introduced in the second Project14 workshop “Vintage Upcycling” by <https://www.element14.com/community/community/project14/hardwarehacking/blog/2020/06/17/radio-magic-sounds-nice-part-1>

Repository link: <https://github.com/alicemirror/RadioMagic>

Radio Magic is the name of a series of three projects completing the upcycling series of this book. This upcycling of a 1960’ Bush Radio names the three projects series; it is a path I started last year to make music with vintage technologies integrated to the most recent hardware and software devices available today.

As already mentioned in this volume, with the upcycling projects I tried to propose ideas, some methodological suggestion and – hopefully – inspiring other makers to hack, change, rebuild and adapt the projects to their vintage devices. Consider this project as a proposal to upcycle an old transistor radio to something new, almost any kind of radio of the same period.

How can an old transistor radio be used to make music?

Electronic music can be created with digital or analogue synthesisers: it is an electronic device able to generate some kinds of a waveform in the audible range (from 20 to 20.000 Hz). There are a lot of functions that can be added to a simple waveform to apply any kinds of effect, chaining together multiple devices or software modules to change the original sound giving accents, modifying decay, duration, attack and much more.

Vintage Upcycling

Indeed, this is not the only kind of approach to electronic music; instead of artificially generated sounds, we can sample them. In many modern synthesisers, the instruments are built starting from a real-world sample. A sound sample is the recording of a short sequence of sounds: a musical instrument, a noisy tool, the sound generated by a percussion, or –why not – a vintage radio.



The author and the radio, a *Blue Baby* special edition of the BUSH transistor radio, 1960 circa.

One of the first music synthesisers was the Mellotron is an electro-mechanical musical instrument developed in Birmingham, England, in 1963. The instrument is played pressing its keys, each of which pushes a length of magnetic tape against a capstan, which pulls it across a playback head. Then as the key is released, the tape is retracted by a spring to its initial position. Different portions of the tape can be played to access different sounds. (<https://en.wikipedia.org/wiki/Mellotron>)

The mid-60' BUSH radio of which I own a Blue Baby limited edition required some upgrade to become a sound generator. The interesting aspect of sampling from a radio is the ability to tune the receiver covering nearby frequencies creating a continuous variation of random sound and noise.

The possibilities offered to create new and original musical instruments with this method are limitless; changing the tuning while sampling we can sample a mix of sounds used as the base to generate an entire set of notes.

5.2 Hacking the Radio Tuner

To hack the tuner to automatically change the radio station I checked how it works inside of the radio.

The radio tuning is controlled by a *variable capacitor*, a widely used component in the radios produced for many decades until the early 70'.

It is not easy operating the right selection rotating the shaft of the variable capacitor. The tuning mechanism has a thin cotton thread connected through a pulley to the manual knob, increase the rotation for more precise identification of the desired band.



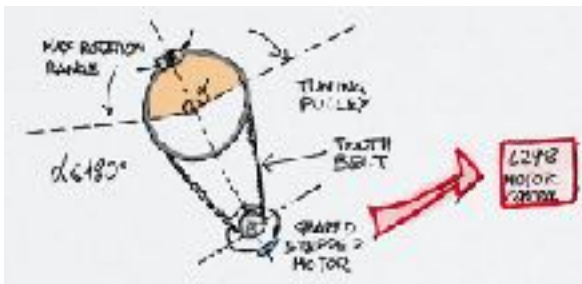
The tuner mechanism: the variable capacitor to the left and the tuner knob to the right.

A variable capacitor is a device whose capacitance may be intentionally changed mechanically or electronically. Variable capacitors are often used in L/C circuits to set the resonance frequency, e.g. to tune a radio (therefore it is sometimes called a tuning capacitor or tuning condenser).

According to the transistor radios technology, the solution described in this project can be easily applied – with some minor adaptations – to a wide range of vintage radios.

The Tuner Mechanism

The variable capacitor has a large pulley driven by the knob pulley through a cotton thread doing a complex path. I wanted to gain direct control to the variable capacitor inverting the functions of the two pulleys.



The variable capacitor rotation angle and the optimal position of the stepper motor.

angle of the shaft; for this reason, I decided to use a geared stepper motor 28BYJ-48 (<https://opencircuit.shop/Product/28BYJ-48-5V-stepper-motor-4-phase-5-wire>), an inexpensive 5V stepper motor.

Due to its reduced dimensions, it is perfect to be added inside of the case of the radio while the reduction gear gives the slow speed and torque needed to move the tuner pulley.

5.3 The New Components

A considerable part of the case is occupied by the back of the big mono speaker of the radio.

Radio Magic

To place the stepper motor shaft in-line with the tuner pulley I had to remove the speaker.

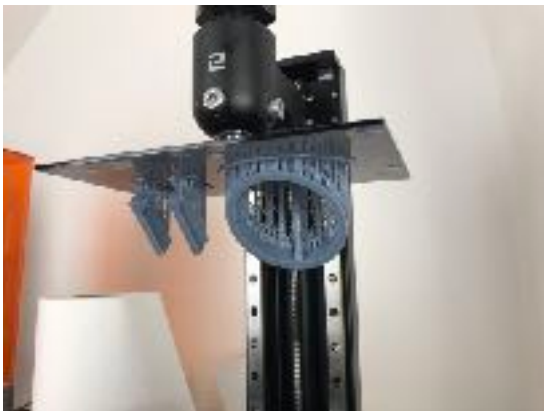
To use the radio as a sampling device the only useful output is the audio output, a 3.5 mm jack on top of the radio. Excluding the amplifier to remove the speaker does not impact the project expectations.

After removing the speaker I checked the available space and the thickness of the case. About 2,5 cm were still available on top of the tuner. I planned to insert the stepper motor and an extra pulley inside this space.

For the movement transmission between the motor and the tuner, I used a 6mm tooth-belt; the stepper motor should be at the same height of the driven pulley for the right movement.



The large speaker on the case top cover has been removed to make space for the stepper motor.



The 3D printed parts (left) and the motor connected through the tooth-belt to the second tuner pulley (right).

Vintage Upcycling

To drive the tuner shaft, I had a limitation, preserving the original pulley to drive the tuner knob while the motor changes the position. The solution has been designing with Fusion360 and 3D with the LCD Elegoo Saturn 3d printer (<https://www.elegoo.com/collections/frontpage/products/elegoo-saturn-msla-4k-monochrome-lcd-3d-printer>).

I had to take care of the right dimensions as the 12 teeth of the stepper motor should be at the same height as the tuner pulley, and the added parts should not be higher than 1.5 cm.

To support the stepper motor, I used two separate supports that can be regulated to give the right tension to the tooth-belt.



The new components assembled and connected to the RJ45 plug on the right side of the radio.

the control board.

The RJ45 cables have eight wires; as the radio needs 9V power (I removed the battery holder for a battery model no longer available), I used the remaining two wires of the Ethernet cable to power the radio itself.

Thanks to a small square hole connecting the battery holder on the back to the radio circuit, it was possible to replace the battery holder with the L298 stepper controller board.

To keep the almost intact the device exterior, I faced the problem of how to connect the six wires needed to control the motor: four for the stepper and two for powering the L298 controller board.

The tiniest solution I found has been using an RJ45 plug: I had to make only a small squared hole on one side of the radio where I hot-glued an Ethernet plug.

With this solution, a single Ethernet patch cable was sufficient to connect the radio to

Vintage Upcycling

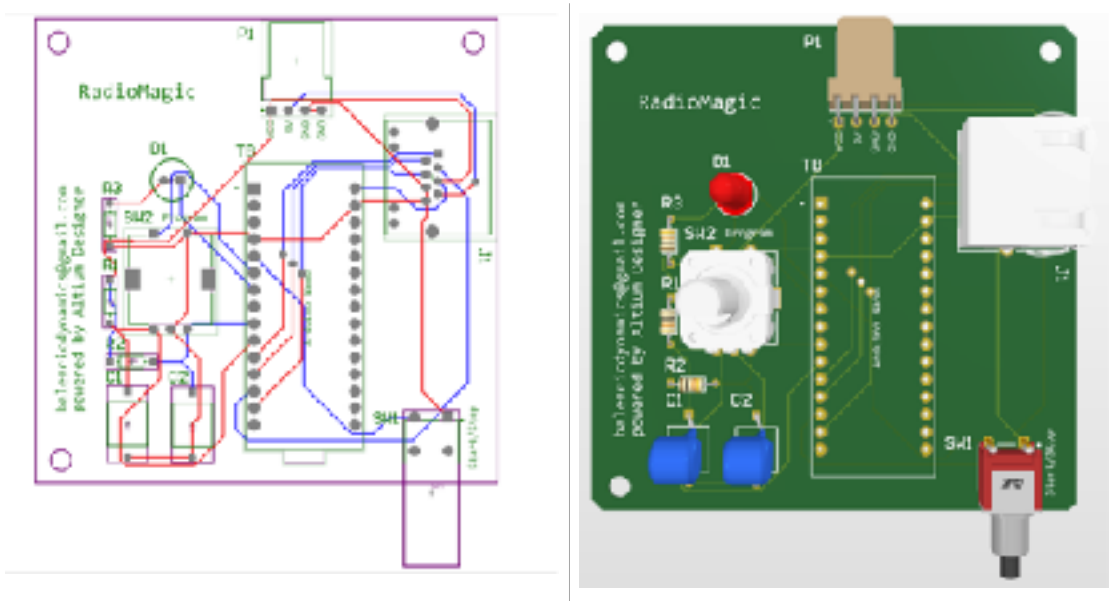
As shown in the schematics, the core part of the board is an Arduino Nano controlling the logic of the tuner. The motor movement and direction are controlled by the Arduino, through a rotary encoder.

Another feature of the board is the ability to program the tuner setting; when the user positions the tuner on the desired frequency, pressing the pushbutton of the rotary encoder a green LED signals the acquired position.

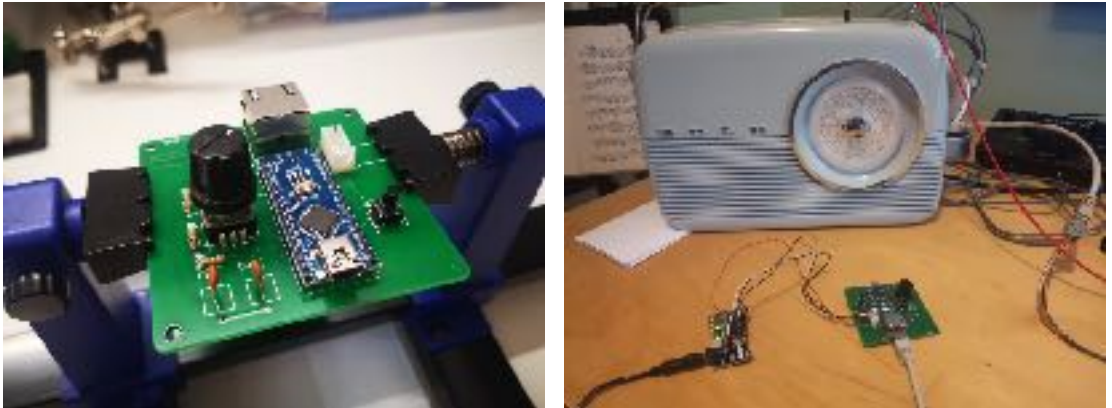
Then the user should rotate the tuner again (no matter the direction) and press the rotary encoder pushbutton another time.

Programming is set and the motor starts rotating the tuner continuously in both directions until the rotary encoder knob is not rotated again. A second pushbutton can be used to temporary stop/start the rotation.

As the feedback of the current position of the tuner is shown accordingly on the radio gauge of the knob – now driven by the tuner – it was not necessary to add a stop switch to the extreme sides of the rotation range.



The control board routed PCB (left) and a 3D simulation of the board (right). Created with *Altium Designer*.



The assembled PCB (left) and the control board connected to the radio during the software development (right)

5.5 The Arduino Sketch

Managing the features seems a simple task but getting the system reactive and fast required some software solutions that are worth to see in detail. The architecture of the software manages asynchronously three components: the notification LED, the stepper motor and the rotary encoder.

The program logic works as a state machine, where every state changes the behaviour of the components. For example, the rotary encoder will disable the programmed rotation when it is moved, while the LED blink different according to the state of the program without blocking the main loop activity.

The snapshot of the program is defined in the RadioStepper structure. It is updated every loop() cycle.

```
/**
 * The RadioStepper structure contains the status of
 * all the parameters controlling the behavior of the radio
 */
struct RadioStepper {
  /**
   * The starting position has been selected
   */
}
```

Vintage Upcycling

```
* This happens when the user press for the first
* time the rotary encoder button. From that point,
* the number of effective steps is counted until
* the button is not pressed for the second time.
*/
bool isSelected = false;

/**
 * When the button has been pressed for the second time, the
 * programmed status indicates that the system is ready to
 * loop the tuner.
 */
bool isProgrammed = false;
/** Status enabled when the tuner is looping
bool isLooping = false;
/** Current relative tuner position inside a loop
int tunerPosition = 0;
/** Looping direction. It is inverted when one of the two limits
/** is reached
int loopDirection = 0;

/**
 * Steps units expressed in number of rotary pulses
 *
 * The units are added algebraically to the counter until
 * the rotary encoder button is not pressed for the second
 * time. At this point the controller is programmed to execute
 * a loop.
 */
int loopSteps = 0;
/** Current rotary encoder position
int16_t encValue = 0;

/** Current LED non-stop blinking frequency
/** It is different when looping is stopped by the stepper is
/** programmed
int blinkLEDFrequency;
/** LED status, inverted during the non-blocking blinking mechanism
boolean isLEDOn = false;
```


Vintage Upcycling

```
    }
}

/**
 * Blink the signal LED for a specified period (ms). If the period duration is
 * less than the frequency needed to blink twice, the function do nothing.
 *
 * \param period The blink duration in ms
 */
void blinkLEDPeriod(int period) {
    // Check that the period is at least four times the blink frequency
    if(period / 2 >= LED_FREQ * 2) {
        //! The number of blinks (On/Off) of the LED
        int stepBlink = period / LED_FREQ;
        //! Blink loop
        int j;
        boolean isOn = true;

#ifdef DEBUG
        Serial << "blinkLEDPeriod(" << period << ") stepBlink " << stepBlink <<
            " Frequency " << LED_FREQ << endl;
#endif

        // Loop for the needed period
        for(j = 0; j < stepBlink; j++) {
            // Invert the last LED status and set the LED
            if(isOn) {
                digitalWrite(PROG_LED, HIGH);
                isOn = false;
            }
            else {
                digitalWrite(PROG_LED, LOW);
                isOn = true;
            }
            delay(LED_FREQ);
#ifdef DEBUG
            Serial << " " << isOn;
#endif
        }
    }
}
```

Radio Magic

```
// Reset the LED to off
digitalWrite(PROG_LED, LOW);
#ifdef DEBUG
  Serial << endl << " END." << endl;
#endif
}
```

The led blinking is controlled through two functions shown above: *blinkLEDPeriod()* that blinks the LED for a predefined period, and *blinkLEDOnce()*. This function is called every *loop()* cycle to invert the last LED status, according to the program state (programming, standby, stepper running, etc.)

The Rotary Encoder

To control the rotary encoder, I have used the *ClickEncode* library (<https://github.com/robogeek78/SparkCore-ClickEncoder>). The advantage of this library is the use of the timer interrupt; so the rotation counter is updated independently by the status of the machine, that is changed accordingly by the interrupt callback function. The main *loop()* function checks the state of the encoder at every cycle as shown below.

```
void loop() {
  // Read the encoder value. Maybe -1, 1 or 0
  radioStepper.encValue = encoder->getValue();

  // Check if the rotary position has changed (exclude the zero status
  if ( (radioStepper.encValue != 0 ) && (encoderCounter == ENCODER_READINGS)) {
    if(radioStepper.isProgrammed == true) {
      // If the tuner is programmed and the user moves the rotary encoder
      // the programmed status is automatically reset
      setProgrammingStatus(false);
    }
    encoderCounter = 0; // Reset the counter readings
    // Check for the direction (clockwise or counterclockwise)
    if (radioStepper.encValue == ROTARY_CW) {
      radioTuner.step(ONE_MOVE_CLOCKWISE);
      // Update the loop counter (only if programming is set)
      updateLoopCount(ONE_MOVE_CLOCKWISE);
    }
  }
}
```

Vintage Upcycling

```
} // Clockwise rotation
else {
    radioTuner.step(ONE_MOVE_COUNTERCLOCKWISE);
    // Update the loop counter (only if programming is set)
    updateLoopCount(ONE_MOVE_COUNTERCLOCKWISE);
} // Counterclockwise rotation
} // Rotary encoder has been moved twice
else {
    if(radioStepper.encValue != 0){
        encoderCounter++;
    }
} // First encoder reading

// Check for the rotary encoder button press. The 0 value shown on power-on can't be
selected
ClickEncoder::Button encButton = encoder->getButton();
if(encButton == ClickEncoder::Clicked) {
#ifdef DEBUG
    Serial << "Encoder button clicked" << endl;
#endif
    if(radioStepper.isSelected == false){
        radioStepper.isSelected = true;
        setProgrammingStatus(false);
        // LED fixed on
        digitalWrite(PROG_LED, HIGH);
#ifdef DEBUG
        Serial << "isSelected true" << endl;
#endif
    } // Button pressed for the first time: start programming the range
    else {
#ifdef DEBUG
        Serial << "Set prog status true" << endl;
#endif
    } // Programming ended, start looping
} // Encoder button clicked

// Check for looping
if(radioStepper.isLooping == true) {
    radioStepper.tunerPosition += (STEPPER_INCREMENT * radioStepper.loopDirection);
```


Radio Magic

```
// Check if the direction should be inverted
if( (radioStepper.tunerPosition == 0) || (radioStepper.tunerPosition ==
radioStepper.loopSteps) ) {
    radioStepper.loopDirection *= -1; // Invert the loop direction
}
radioTuner.step(STEPPER_INCREMENT * radioStepper.loopDirection);
}

// Non-blocking LED blinking, if needed
blinkLEDOnce();
}
```

There are three status events modified by the rotary encoder: the encoder button pressed, the rotation of the encoder while programming a motion range of the tuner, and the rotation of the encoder when the tuner is programmed.

The Run/Standby Pushbutton

Also, this component acts independently modifying the status of the running program: if the stepper is programmed and the tuner is moving the motor is stopped, and the LED blinks slower; when it is pressed again, the motor restart the rotation sequence.

If the pushbutton is pressed when the stepper motor is not programmed, it does not affect.

The Run/Standby pushbutton is connected the Arduino Nano pin 2 corresponding to the hardware interrupt IRQ_0.

When an interrupt occurs on this pin the callback function *irqLoopButton()* is called automatically regardless of the main loop() execution status.

```
/**
 * IRQ Vector callback for Nano IRQ 0 (the loop control button pin)
 */
void irqLoopButton() {
    if(radioStepper.isProgrammed == true) {
        detachInterrupt(digitalPinToInterrupt(LOOPER_BUTTON));
        // If the tuner is programmed, change the status of the loop flag
        if(radioStepper.isLooping == true) {
            radioStepper.isLooping = false;
        }
    }
}
```

Vintage Upcycling

```
        radioStepper.blinkLEDFrequency = LED_IDLE;
#ifdef DEBUG
        Serial << "LED idle" << endl;
#endif
    } else {
        radioStepper.isLooping = true;
        radioStepper.blinkLEDFrequency = LED_FREQ;
#ifdef DEBUG
        Serial << "LED frequency" << endl;
#endif
    }
    attachInterrupt(digitalPinToInterrupt(LOOPER_BUTTON), irqLoopButton, LOW);
    delay(10);
    radioStepper.millisCounter = millis();
}
}
```


Finished to write, November 10th, 2020

The Author



Enrico Miglino, developer, maker and digital creator, upcycler. AI and image analysis expert, he has been an active member of the Makers movement for twenty years. As a blogger and a technical writer, he collaborates with several international magazines. His works, a combination of electronics, mechanics and software, have been exhibited at several international events. Besides, he strongly supports the Open Source community and participates in STEM projects.

From the 1970's design desk lamp up to the *Radio Magic* projects, an inspiring guide on vintage upcycling for makers. The projects are based on the articles published by the author on *The Shed* magazine and award-winning projects from the *Element14.com* Project14 challenges.

This is an inspiring work for vintage lovers and makers.

All the resources mentioned in this book, from the software sources to the 3D STL files, circuit schematics, and development strategies, are available on the projects GitHub repositories under Open Source LGPL and CC 3.0 licenses.

Through the book projects you will discover:

- How to design an animatronic
- How to transform a radio into a musical instrument
- How to use microcontrollers and single-board computers to change old devices behaviour
- How to make an analog musical synthesizer

And more...

Preface by Greg Vincent, Publishing Editor *The Shed Magazine*.

Introduction by Tariq Ahmad, Community Manager of Project14 at *element14.com*.

The projects in this book have been created thanks to the support of the following sponsors:

