The latest incarnation of a DES Day Counter was implemented in March 2025.
See: <http://darcy.rsgc.on.ca/ACES/Projects/DayCounter/index.html>
The code below supports the DES breadboard prototypes developed for the project as an introduction to RF to ICS3U-E.

Transmit

// PROJECT  :TxMasterRotaryEncoder

// PURPOSE  :Day Counter Master Transmitter

// DEVICE   :Genuine Nano + nRF24L01 + Rotary Encoder

// AUTHOR   :C. D'Arcy

// DATE     :2025 03 16-...

// uC       :328p (Nano)

// COURSE   :\*

// STATUS   :Working.Paired with Large7SegmentReceiver.ino or RxSlave.ino

//          :It was the knockoff Nano that couldn't handle the power.

//          :When I switched to the Genuine Nano, it worked.

// NOTES    :Can be finicky...

#include <SPI.h>

#include <RF24.h>

RF24 radio(9, 10);  //CE (Chip Enable/Disable), CSN(SS)

byte addresses[][6] = { "DESDC", "2Node" };

#define CHANNEL 91  //Any value between 0 and 125 (2.400GHz to 2.525GHz)

#define DEBUG true  //Conditional use of Serial support

#define RE\_PINA 2   // A

#define RE\_PINB 3   // B

boolean CW = true;  //direction: CW or CCW (not CW)

volatile boolean triggered = false;

uint8\_t day = 0;  // self-explanatory

void setup() {

  if (DEBUG) {

    Serial.begin(9600);

    while (!Serial);

  }

  radio.begin();                  //invoke the radio object

  radio.setPALevel(RF24\_PA\_MIN);  //close range so minimum power sufficient

  radio.setChannel(CHANNEL);      //Tx and Rx communication on same channel

  radio.openWritingPipe(addresses[0]);  //Transmit assumes these pipes

  radio.stopListening();

  attachInterrupt(digitalPinToInterrupt(RE\_PINA), ISR\_Rotary, RISING);

}

void loop() {

  if (triggered) {

    triggered = false;

    //1. First, determine the direction (hardware (logic) strategy)...

    CW = digitalRead(RE\_PINA) ^ digitalRead(RE\_PINB);

    //2. Now, determine whether to decrement or increment the day number

    if (CW)

      day = (day + 1) % 10;

    else

      day = (day == 0) ? 9 : day - 1;

    if (DEBUG) {  //Conditional Serial support for debugging purposes

      Serial.println(CW);

      displayStates(day);

    }

    //3. Finally, Transmit...

    radio.write(&day, 1);  //transmit day number, one byte

  }

}

void ISR\_Rotary() {  //Keep ISR bodies as short as possible

  triggered = true;

}

//for Debugging purposes, if necessary...

void displayStates(uint8\_t dayNumber) {

  Serial.print(dayNumber);

  Serial.print(',');

  uint8\_t stateA = digitalRead(RE\_PINA);

  uint8\_t stateB = digitalRead(RE\_PINB);

  Serial.print(stateA);

  Serial.print(',');

  Serial.println(stateB);

}



# Receive

// PROJECT  :RxSlave

// PURPOSE  :Day Counter Slave Receiver (for TxMasterRotaryEncoder.ino)

// DEVICE   :Nano + nRF24L01 + 74HC595/6 + Big 7-Segment Display

// AUTHOR   :C. D'Arcy

// DATE     :2025 03 16-...

// uC       :328p (Nano)

// COURSE   :\*

// STATUS   :Working (when used with TxMasterRotaryEncoder.ino)

//     :http://darcy.rsgc.on.ca/ACES/TEI3M/CommunicationProtocols.html#RF1

//     :https://tmrh20.github.io/RF24/classRF24.html

//     :http://darcy.rsgc.on.ca/ACES/Datasheets/nRF24L01.pdf

#include <SPI.h>

#include <RF24.h>

#define CHANNEL 91 //any value between 0 and 125 (2.400GHz to 2.525GHz)

uint8\_t receivedData; //for receiver to store the incoming byte

RF24 radio(9, 10);    //CE (Chip Enable/Disable), CSN(SS)

uint8\_t addresses[][6] = { "DESDC", "2Node" };//pipe names

#define DEBUG false  //Conditional use of Serial support

#define DATA 7       //74HC595/TPIC6C596

#define CLOCK 6      //

#define LATCH 5      //

#define RE\_PINA 2    // possibly nRF24L01: IRQ at some point

#define RE\_PINB 3    // ?

// Developed on Small 0.56" 7-Segment Display

uint8\_t segments[] = {

  //abcdefg0

  B11111100, B01100000, B11011010, B11110010,  //0..3

  B01100110, B10110110, B10111110, B11100000,  //4..7

  B11111110, B11100110                         //8..9

};

// Live on Large 6.5" Sparkfun 7-Segment Display (COM-08530)

uint8\_t Segments[] = {

  //abcdefg0

  B11111100, B01100000, B11011010, B11110010,  //0..3

  B01100110, B10110110, B10111110, B11100000,  //4..7

  B11111110, B11100110                         //8..9

};

uint8\_t numSegments = sizeof(segments);

volatile boolean triggered = false;

uint8\_t day = 1;

void setup() {

  if (DEBUG) {

    Serial.begin(9600);

    while (!Serial)

      ;

  }

  //prepare shift register control pins

  pinMode(DATA, OUTPUT);

  pinMode(CLOCK, OUTPUT);

  pinMode(LATCH, OUTPUT);

  // Test shiftout...

  digitalWrite(LATCH, LOW);

  shiftOut(DATA, CLOCK, LSBFIRST, segments[day]);

  digitalWrite(LATCH, HIGH);

  // (Not Implemented yet) External interrupt signalling new RF data...

  attachInterrupt(digitalPinToInterrupt(RE\_PINA), ISR\_nRFIRQ, CHANGE);

  radio.begin();                  //invoke the radio

  radio.setPALevel(RF24\_PA\_MIN);  //low power ampl. for close proximity

  radio.setChannel(CHANNEL);      //mutually agreeable frequency

  radio.openWritingPipe(addresses[1]); //select ONE addresses to write to

  radio.openReadingPipe(1, addresses[0]);//Rx and Tx are reversed

  radio.startListening();                  //should be good to go...

  if (DEBUG)

    displayAll();

}

void loop() {

  if (radio.available()) {  //anything to be read?

    radio.read(&day, 1);    //grab the day number [0..9]

    digitalWrite(LATCH, LOW);

    shiftOut(DATA, CLOCK, LSBFIRST, segments[day]);

    digitalWrite(LATCH, HIGH);

    if (DEBUG)

      Serial.println(day);  //local echo

  }

  //if and when IRQ interrupt enabled...

  if (triggered) {

    triggered = false;

    digitalWrite(LATCH, LOW);

    shiftOut(DATA, CLOCK, LSBFIRST, segments[day]);

    digitalWrite(LATCH, HIGH);

  }

}

void ISR\_nRFIRQ() {  //Keep ISR bodies as short as possible

  triggered = true;

}

//for Debugging purposes if necessary...

void displayAll() {

  for (uint8\_t d = 0; d < numSegments; d++) {

    digitalWrite(LATCH, LOW);

    shiftOut(DATA, CLOCK, LSBFIRST, segments[d]);

    digitalWrite(LATCH, HIGH);

    Serial.println(d);

    delay(1000);

  }

}

