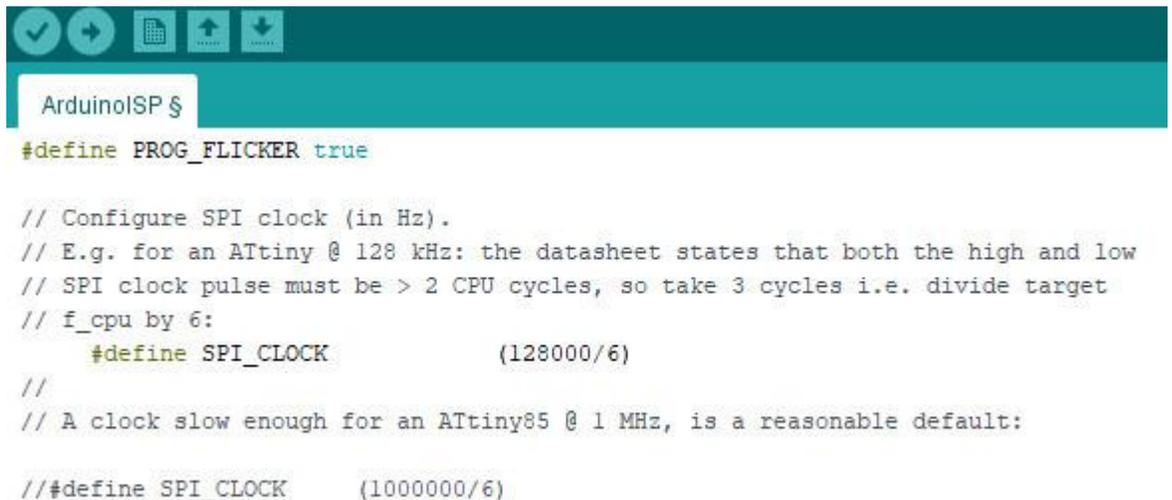


ATTiny Programming for the Mini DIN sequencer using an Arduino Nano

As the ATtiny doesn't have a communications port programming in with the Arduino IDE is a bit more complicated than simply plugging in a USB connector. The In Circuit Serial Programming (ICSP) port will be used for programming.

1. Program the Arduino Nano as an ISP programmer
 - a. Attach a Nano to the computer via USB cable. Do NOT connect the Nano to any other circuitry at this point or you'll likely get programming errors.
 - b. Download and / or open the Arduino IDE. The Download can be found here: <https://www.arduino.cc/en/software>
 - c. Navigate to 'File' then 'Examples' then 'ArduinoISP' then move the mouse right to reveal 'ArduinoISP' option and select it to load the program.
 - i. As the wiring to the ATtiny is a bit of a rats nest we will modify the 'ArduinoISP' script to use a lower programming speed, i.e. 128k. This speed will also work when programming a device for any clock speed, it simply slows down the programming rate and may be useful with the additional lead lengths needed to connect the sequencer to the programmer.
 - ii. Navigate to the 'Configure SPI clock (in Hz.)' section of the sketch. Uncomment the line `#define SPI_CLOCK (128000/6)`
 - iii. Comment out the line `#define SPI_CLOCK (1000000/6)`
 - iv. This changes the programming rate from 1MHz to 128 kHz



```
ArduinoISP §
#define PROG_FLICKER true

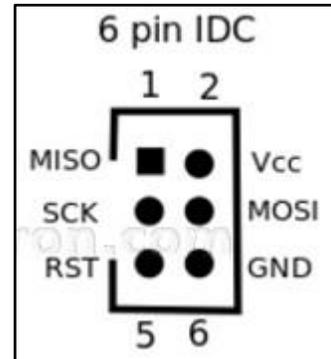
// Configure SPI clock (in Hz).
// E.g. for an ATtiny @ 128 kHz: the datasheet states that both the high and low
// SPI clock pulse must be > 2 CPU cycles, so take 3 cycles i.e. divide target
// f_cpu by 6:
    #define SPI_CLOCK          (128000/6)
//
// A clock slow enough for an ATtiny85 @ 1 MHz, is a reasonable default:

// #define SPI_CLOCK          (1000000/6)
```

- d. Navigate to the 'Tools' menu then 'Board:' then 'Arduino AVR boards' and select 'Arduino Nano'
- e. Again navigate to the 'Tools' menu then 'Processor' then select 'ATmega328P'
Note if this fails to program correctly you may have an older boot loader in

which case go back to the 'Tools' 'Processor' and select 'ATmega328P (Old Bootloader).

- f. Again navigate to the 'Tools' menu then 'Port:' and make sure the correct com port is selected.
 - g. Program the Nano. Navigate to the right arrow in a circle icon near the top of the screen and select it or navigate to the 'Sketch' menu item then select 'Upload'
2. Put together a programming jig. Solder pads are provided on the sequencer PCB for wires that can be connected as indicated in the table below. Keep the wire lengths under about 10 cm. It's convenient to use a solderless breadboard as a host for the Nano and wires but not absolutely necessary. The Nano ICSP header can also be used instead of the D pins with the exception of the reset pin which must be connected to D10 on the Nano. Don't forget to add a 10 uF capacitor from RST (+) to Gnd (-) on the Nano.



Function	ATtiny physical pin	Sequencer Solder Pad	Nano Pin	Nano ICSP 6 pin header
Reset	1	SP7	D10	D10 (not on ISP)
MOSI	5	SP6	D11	4
MISO	6	SP2 (DIN Pin2)	D12	1
SCK	7	SP8 (DIN Pin8)	D13	3
V+	8	SP1 (DIN Pin1)	5V	2
GND	4	SP11 or SP12	GND	6
10uF cap			RST to Gnd	

3. Prepare the ATTiny 45 for programming with a bootloader
 - a. Download the ATTinyCore to add these boards to the boards manager
 - i. Click on 'File' then 'Settings'
 - ii. In the 'Additional Boards Manager URLs:' section near the bottom of the window paste the following link
 - iii. http://drazzy.com/package_drazzy.com_index.json
 - iv. Click on 'OK'
 - b. Click on 'Tools' then 'Board: 'Arduino Nano', then right and click on 'Boards Manager'
 - c. In the 'Filter your search...' at the top of the window type in 'ATTinyCore'
 - d. Click on the 'ATTinyCore window area.
 - e. Close the 'Boards Manager' window.
 - f. Click on 'Tools' then 'Board: ...' and navigate to 'ATTiny Core'

- g. Select 'ATtiny 25/45/85'
 - h. Click on 'Tools' then 'Chip: ...' then navigate to and select 'ATtiny45'
 - i. Click on 'Tools' then 'Clock: ...' then navigate to '1MHz internal'
 - j. Leave other menu items in the 'Tools' section in their defaults.
 - k. Click on 'Tools' then 'Port' and check that the correct port is still selected.
 - l. Click on 'Tools' then 'Programmer'. Navigate to and select 'Arduino as ISP (ATTinyCore)'
4. Load the Mini-DIN sequencer sketch
 - a. Click on 'File' then 'Open'
 - b. Navigate to and select the 'Sequencer for Ver2.2 hw_V2.2a.ino'
 - c. Make any desired changes to the StartDelay or Hangtime but if you do make changes save the file with a different name.
5. Add elapsedMillis to the Arduino library
 - a. Under the 'Tools' menu click on 'Manage Libraries'
 - b. In the library manager search for 'elapsedMillis' and install the latest version
6. Click on the 'Tools' menu and check that all the options for board, chip, clock speed, and port are still valid.
7. Burn the Bootloader and then the sequencer sketch
 - a. Click on the 'Tools' menu then select 'Burn Bootloader'
 - b. Upload the sketch using the right arrow in the circle in a menu near the top or click on 'Sketch' then 'Upload'
8. Unsolder all the programming leads from the sequencer and try it out!