Text file saved to notepad

The Universal Serial Interface supplied with some of the ATtiny devices offers many options for communication with other devices and with a PC.

When communicating with a PC the USI is configured to behave like a UART (the more usually supplied receiver transmitter block).

Three means of clocking the USI are provided. A timer interrupt, and external clock or a software generated clock.

The interrupt system enables two processes to run at the same time. Data transfer plus an underlying process that can be interrupted. Setting up the interrupts does however take a significant amount of time which limits the maximum speed at which data transfer can take place.

The software generated clock simply waits for the clock pulse. No underlying process is possible but the maximum possible data speed is significantly higher.

The external clock option probably offers the best of both world but has not been tested for this project.

For many of us sending strings and user prompts to the PC and receiving individual keypresses from the keyboard, the SW clock may be the best option.

An additional advantage is that in the absence of latency due to the interrupt the clock period can be set up simply and except for the highest speeds (~256000B) nearly always conforms with simple calculation.

Freaks Intro

Simple USI configuration provides UART function

There are many applications in which the USI () is only required to send strings and user prompts to a PC and receive keypresses from the keyboard. The user requires:

A straightforward method of deriving the USI clock signal:
    No issues due to interrupt latency
    Simple calculation of the clock period
The ability to use any popular baud rate: typically 115200 and above.
Minimum hardware: No external clock source
    Possibly use of the default RC clock

The user does not require:
To have an underlying process that must not be halted while the USI is functioning.
The requirement to determine and take into account any interrupt latencies.

USI hardware provides for all these requirements, both simple and more demanding. Any one of three clocks sources can be selected:
    Timer interrupts which enable the presence of an underlying processes.
    Soft ware interrupts which provide the basic configuration considered here.
    An external clock for the most demanding applications.

User documentation

A project is supplied that enables this software (SW) USI implementation to be tested. This program can be used as a template of others to employ or they can use copy and paste to retrieve bits of interest.

It has been tested on the ATtiny 461, 861 and ATtiny 44, 84 devices which provide 14 and 20 pin hardware with 4kB or 8kB of program memory.

Recommended steps for getting things going are:

Open project UNO_AVR_programmer_V2 posted here on 21 Oct 2020.
Download
UNO_AVR_programmer_V2
Cal_ATtiny_44_461_V2
Use Arduino to upload the programmer to the UNO development board
Calibrate the target as described in the project documentation

Download Project “High_speed_USI” from here.
Use the UNO_AVR programmer to upload the High speed USI to a target.

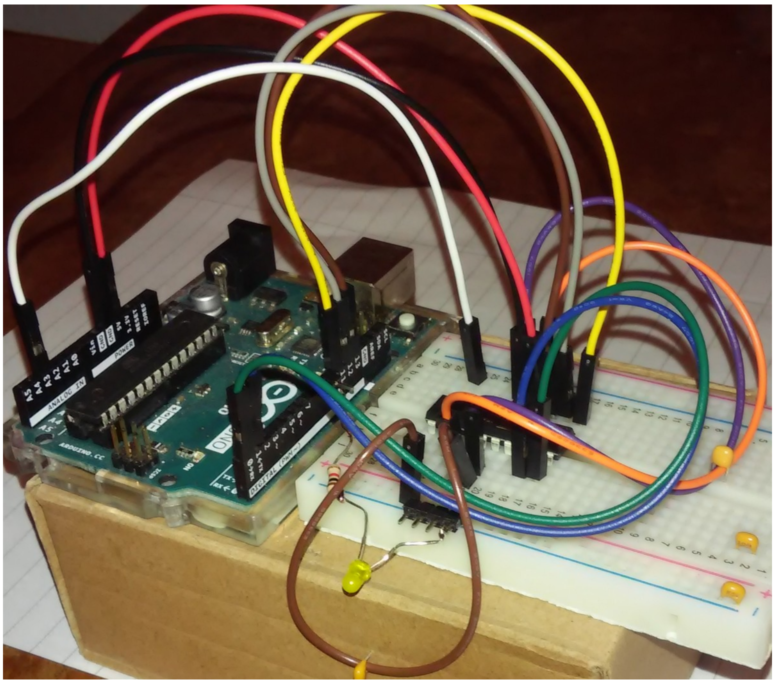
When programming is complete the target should automatically run. It generates user prompt “R…..R…..R……..”.
Press “r” to run a series of USI tests or “R” to read a text file saved to flash.

Note: The Project.h file requires user input:

The baud rate can be selected (options have been set up for 9600, 19200, 38400, 57600, 76800, 115200, 128000, 144000 and 256000 Baud).

There are two sections, one for ATtiny 44/84 devices and one for 461/861 devices. The unwanted section should be commented out.

Two sample text files are provided to be saved to program memory. One is suitable for 4K devices and the other for 8K.

	**Photo of the development setup for the ATtiny 461/861** Wires are RS part no 791-6463 Breadboard is RS part no 102-9147 5V Decoupling has not been used with the ATtiny

Interconnections

Wire Colour	UNO		ATtiny 461/861	
				Pin
Red	5V		+5	5
Black	GND		0V	6
White	A3	PORTC3	PB7 (RESET)	10
Yellow	Digital IO 11	MOSI	MOSI PB0	1
Grey	Digital IO 12	MISO	MISO PB1	2
Brown	Digital IO 13	SCK	SCL PB2	3
Blue	Digital IO 1	Tx	USI DO PA1	19
Green	Digital IO 0	Rx	USI DI PA1	20
Orange 2			5V AVCC	Pins 5 to 15
Brown 2			AGND	Pins 6 to 16
Brown 3	Digital IO 8	LED + Res	AGND	Pin 16