ATtiny13A 클럭생성 코딩

# 동기부여

실시간 시계를 ATtiny13A칩을 이용해서 저렴하게 만드는 방법을 상상하면서, 자료를 찾고 오실로스코프로 테스트 하면서 시계를 저렴하게 만들어 본다.

# RTC 코딩 교육 목표

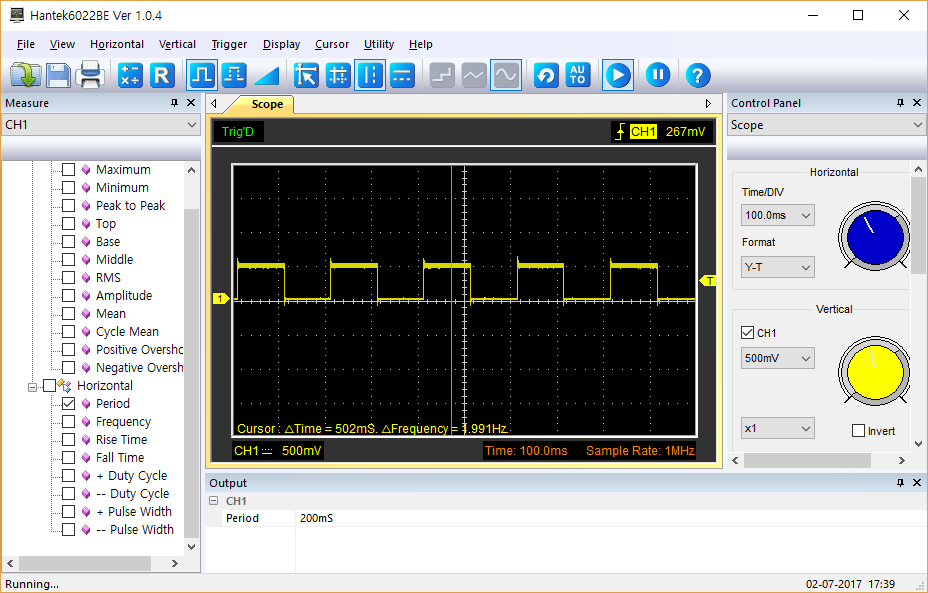
RTC에 대한 이해

AVR MCU의 내부클럭에 대한 이해

외부클럭 또는 RTC모듈없이 꾸며 보는 실시간 시계

# 예상 결과물

아래와 같이 클럭신호(구형파, 펄스트레인)가 발생한다.



# 준비물

ATtiny13A 개발 보드, 오실로스코프(Hantek 6022BE)

# 아두이노 스케치 코드

## ATtiny85 (at 8Mhz)용 코드

테스트해 보니 ATtiny13A (at 9.6Mhz)에서는 340Khz로 동작한다)

// Attiny85 , running @ 8MHZ

// output of 153600 Khz = 9600 baud \* 16

// Square wave output on pin 5 (PB0)

// ATMEL ATTINY85 / ARDUINO IDE

void setup()

{

// OSCCAL calibrates 8 mhz internal timer speed

// value of 96 was used to get output of 153600 Khz at

// pin 5 on an osciilscope for this particular project

OSCCAL = 96; // higher value = faster (see Figure 22-42 in data sheet)

CLKPR = 0x80; // set system clock to 8mhz with no prescale

CLKPR = 0x00; // these 2 CLKPR instructions have to be run together in order

// to set clock to 8 Mhz

DDRB |= (1<<PB0); //Set pin PB0 as output (pinMode(0, OUTPUT)

TCNT0 = 0; // initialize timer counter value to 0

TCCR0A = 0; // write 0 to timer 0 control registers

TCCR0B = 0;

TCCR0A |= (1 << COM0A0); //Timer0 in toggle mode (Table 11-2 of datasheet)

TCCR0A |= (1 << WGM01); //Start timer 1 in CTC mode (Table 11-5 )

TCCR0B |= (1 << CS00); // Prescaler table 11.6

OCR0A=27; //CTC Compare value (27 gives 153 khz)

}//setup

void loop()

{

}

## ATtiny13 (at 9.6Mhz)용 코드

/\*

Target MCU: Attiny13A , running @ 9.6MHZ

Used IDE: Arduino Sketch 1.0.6

Author: Insoo Kim ( insoo@hotmail.com )

Date Created: Sunday, July 02, 2017

Last Updated:

Purpose: To make real-time clock with minimal resource, meaning no use of external clock and RTC module.

Output:

Clock pulse(Square wave) on pin 5 (PB0)

Ref:

https://www.instructables.com/answers/How-to-get-an-ATtiny-to-run-an-accurate-clock/

output of 153600 hz = 9600 baud \* 16

\*/

void setup()

{

// OSCCAL calibrates 8 mhz internal timer speed

// value of 96 was used to get output of 153600 Khz at

// pin 5 on an osciilscope for this particular project

OSCCAL = 0x00; // 158 Khz

//OSCCAL = 0x3F; // 247 Khz

//OSCCAL = 0x7F; // 542 Khz

//CLKPR = 0x80; // set system clock to 8mhz with no prescale

CLKPR = 0x08; // set system clock to 8mhz with prescale 256

CLKPR = 0x00; // these 2 CLKPR instructions have to be run together in order

// to set clock to 8 Mhz

DDRB |= (1<<PB0); //Set pin PB0 as output (pinMode(0, OUTPUT)

TCNT0 = 0; // initialize timer counter value to 0

TCCR0A = 0; // write 0 to timer 0 control registers

TCCR0B = 0;

TCCR0A |= (1 << COM0A0); //Timer0 in toggle mode (Table 11-2 of datasheet)

TCCR0A |= (1 << WGM01); //Start timer 1 in CTC mode (Table 11-5 )

TCCR0B |= (1 << CS02); // Prescaler table 11.9

TCCR0B |= (1 << CS00); // Prescaler table 11.9 by prescale 1024

OCR0A=106; //T = 200ms at PB0, measured by OSC

}//setup

void loop()

{}//loop

# 수질환경 학습참조

## 세계의 수질 문제

## 전해질과 도전 원리응용 수질 측정

## 수자원 고갈 문제와 대책

# 참조

## 기초 자료 (Arduino UNO)

<http://www.instructables.com/id/Arduino-Water-Pollution-Monitor/>

## 심화 자료 (Arduino Mega)

<https://create.arduino.cc/projecthub/eani/water-quality-monitoring-and-notification-system-f85d23>

## 온라인 모니터링 자료 (MS Azure)

<https://create.arduino.cc/projecthub/chanhj/water-quality-monitoring-system-ddcb43>

## 세계 수질 문제

<https://en.wikipedia.org/wiki/Water_pollution>

## 전해질과 도전 원리응용 수질 측정

<https://app.box.com/s/9m2p0qi9o88jhrdcxtg0h3zvy676l2ce>

<http://www.libelium.com/smart-water-sensors-to-monitor-water-quality-in-rivers-lakes-and-the-sea/>

<http://envirotechi.blogspot.kr/2012/08/monitoring-and-detection-methods-of.html>

<http://www.fondriest.com/environmental-measurements/parameters/water-quality/conductivity-salinity-tds/>

<http://www.lenntech.com/applications/ultrapure/conductivity/water-conductivity.htm>

## 수질측정 데이터의 핸드폰망을 통한 문자 전송

<https://publiclab.org/wiki/water-quality-sensor>

## 수자원 고갈 문제와 대책

<https://en.wikipedia.org/wiki/Water_resources>