

# Raspberry Pi制御のCANON RC1互換 リモートコントローラの製作

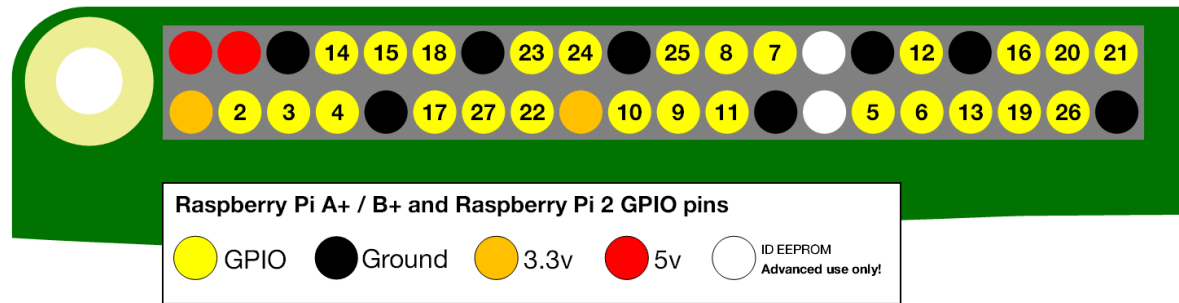
2017/06/13 T.YAsui

# 目的

- photosrvサーバに組み合わせる、CANON RC1互換の赤外線リモコンを製作します
  - コントローラには、旧ATMEL社のATTiny85-20PUを使用します
  - 制御パルスは手元にあったRC6リモコンに準拠させました(CANON設計ノミナル値ではありません)
  - 赤外線パルスのキャリアは8bit Timer/Counter1 で生成しています
  - 参考(先輩方の記事)
    - Canon RC-1 IR remote control reverse-engineered [http://www.doc-diy.net/photo/rc-1\\_hacked/](http://www.doc-diy.net/photo/rc-1_hacked/)
    - Tiny10で実用品を作る その3 デジカメ用赤外線リモコン <http://morecatlab.akiba.coocan.jp/lab/index.php/2012/03/tiny10-ir-remote/>
    - キヤノンのカメラのリモートリリース (またはリモコンリリース) <https://lowreal.net/2013/11/17/1>

# 回路図

## ■ こんな回路を製作してちょ



<https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/README.md>

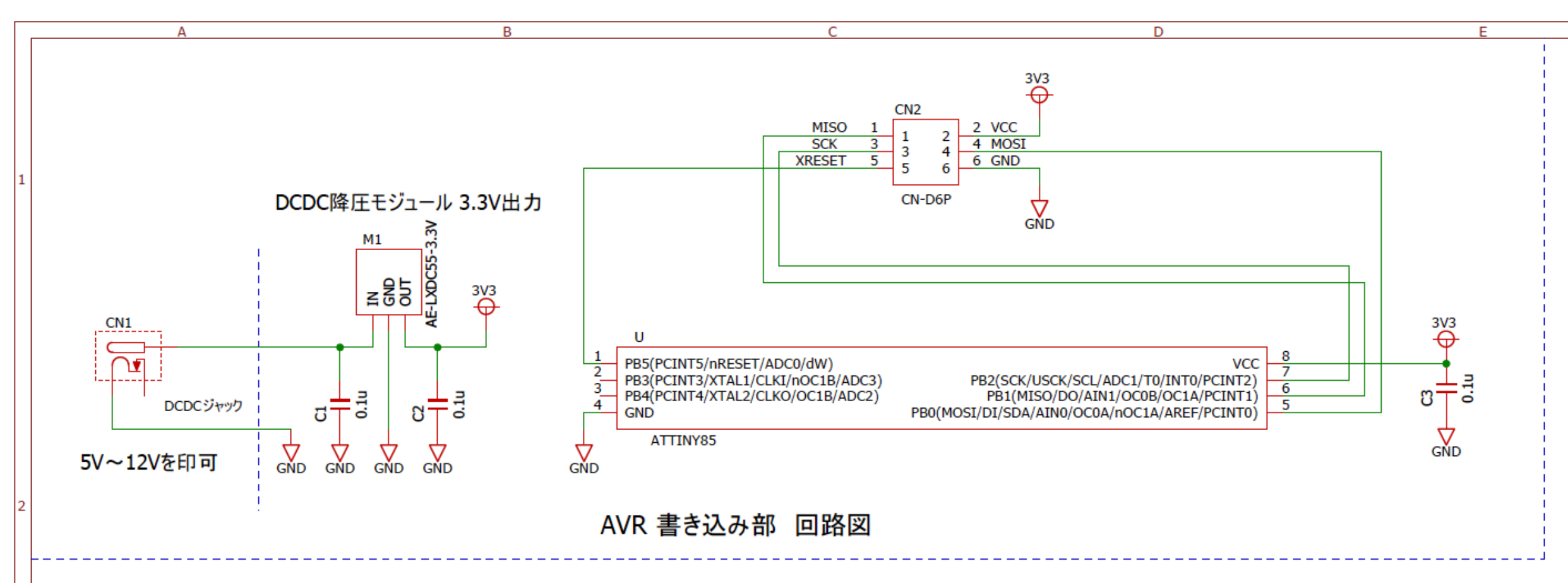
# 回路図

## ■ 設計

- XTRIG入力が、Lに変化した時点で、遅延なしシャッターパルスを発光
- STAT出力はシャッターパルス発光から適当待ち合わせ時間経過までHを維持
- XRESET入力 LでMCUをリセット(トチ狂ったらLを100msかましてください)

# 回路図

## 書き込み回路こんな感じ



AVR 書き込み部 回路図

# ファームウェア

## ・ ATMEL Studio7でビルドして、書き込みしてちょ

```
/*
 * RC1_test1.c ; CANON EOS RC-1 Remote Control test Program
 * 2017/05/21 Rev.0.1 T.YAsui 新規作成
 *
 * (c) COPYRIGHT TYLAB T.YAsui 2017
 */
#ifndef F_CPU
#define F_CPU 8000000UL
#endif

#include <avr/io.h>
#include <avr/sleep.h>
#include <util/delay.h>
#include <avr/interrupt.h>

/* Fuse Bit */
/* Extended Byte = 0xFF(Default) */
/* High Byte = 0b1101 1110 = 0xDE BODLEVEL[2:0] = 110(2.0-1.7V) */
/* Low Byte = 0xE2 bit7 : CKDIV8 =1 */

/* CKSEL[3:0] = 0b0010 : Calibrated Internal Oscillator 8.0MHz */
/* SUT[1:0] = 0b10 : Slowly rising power */

/* 8bit Timer1 define @RC 8MHz center */
#define FREQ_40k 100 /* 40.0kHz */
#define FREQ_38k 106 /* 38.0kHz */
#define FREQ_32k 122 /* 32.768kHz */

#define OC1B_Enable 0b00010000
/* bit6 : PWM1B = 0
   bit5 : COM1B1 = 0
   bit4 : COM1B0 = 1 toggle OC1B
   bit3 : FOC1B = 0
   bit2 : FOC1A = 0
   bit1 : PSR1 = 0
*/
#define OC1B_Disable 0b00000000
```

FUSE Bitの書き換えをお忘れなく

```
/* Port control define */
#define STAT 0x02
#define STAT_H PORTB |= STAT
#define STAT_L PORTB &= ~STAT

#define XTRIG_PINB & 0x01
#define XTRIG_ENB (XTRIG) == 0x00
#define XTRIG_DIS (XTRIG) == 0x01

void RC1_shutter(void) {
    OCR1C = FREQ_32k;

    GTCCR = OC1B_Enable;
    _delay_us(500);
    GTCCR = OC1B_Disable;

    _delay_ms(7);
    _delay_us(330);

    GTCCR = OC1B_Enable;
    _delay_us(500);
    GTCCR = OC1B_Disable;
}

EMPTY_INTERRUPT(PCINT0_vect);

int main(void) {
    /* MCU Init. */
    /* PIN Config
     * PB0 IN : XTRIG(PCINT0) wPU
     * PB1 OUT : STAT
     * PB2 IN : none wPU
     * PB3 OUT : none
     * PB4 OUT : IrLED (OC1B)
     * PB5 IN : nRESET
     */
    MCUCR = 0x00; /* bit6 : PUD = 0 (Pull-up not Disable) */
```

```
PORTB = 0b00000101;
DDRB = 0b00011010;

/* 8bit Timer/Counter1 Init. */
/* Compare Match clear Mode Config */
OCR1C = FREQ_32k;
GTCCR = OC1B_Disable;
TCCR1 = 0b10000001; /* bit7 : CTC1 = 1 (Clear Compare Match)
                    bit6 : PWM1A = 0
                    bit5 : COM1A1 = 0
                    bit4 : COM1A0 = 0
                    bit[3:0] : CS13-10 = 0001(Async mode=CK) */

/* Interrupt Enable */
PCMSK = 0x01; /* bit0 : PCINT0 = 1 enable */
GIMSK = 0x20; /* bit5 : PCIF = 1 enable */
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sei();

/* ----- Main Loop ----- */
while(1) {
    sleep_mode(); /* enter sleep mode */

    if(XTRIG_ENB) {
        STAT_H;

        RC1_shutter();
        _delay_ms(100);

        STAT_L;
    }
}
}
```

# RaspberryPiのGPIO制御コード

## ■ /usr/lib/cgi-bin 配下に配置してね

```
#!/usr/bin/python
# 2017/06/10 Rev.0.1 T.YAsui init. Ver.
# COPYRIGHT TYLAB T.YAsui 2017
import RPi.GPIO as GPIO
import time
```

```
#GPIO init.
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(2, GPIO.OUT)
GPIO.setup(3, GPIO.OUT)
GPIO.setup(4, GPIO.IN)
```

```
GPIO.output(2, 1)
GPIO.output(3, 1)
```

```
# GPIO Control for Camera Shutter
if GPIO.input(4) == 0:
    GPIO.output(3, 0)
    time.sleep(1)
    GPIO.output(3, 1)
```

```
# CGI output
print("Content-type:text/html\n")
print("<html>")
print("<head>")
```

```
print('<meta http-equiv="refresh" content="0;URL=/photosrv/index.html">')
print("</head>")
print('<body>')print('<a href="javascript:history.back()">back</a>')
print("</body>")
print("</html>")
```