

Erkennung S0-Schnittstellen-Impuls

SkobyMobil 1 November 6, 2015, 10:19am

Guuten Tach, Jung's!

Euch geht's noch allen gut? Alles im grünen Bereich und reichlich Erfahrungen mit dem Arduino gesammelt?

Rabenaugen, was macht Deine GPS-Karre? Findet sie jetzt den kürzesten und genauesten Weg nach Hause?

Bin ja lange nicht mehr dabei gewesen und jetzt verzweifel ich an etwas leichten. Bei mir will der Groschen nicht fallen...

Ich habe hier so meine Probleme mit dem erkennen eines S0-Impuls am Mega 2560.

Ich habe nachfolgenden Sketch, der die Impulse von 8-S0-WSZ empfängt.

Das funktioniert bis jetzt recht zuverlässig und genau...

```
/** Hilfsprogramm zur Auswertung der Zähler
 * Maximal 8 S0-Zähler können angeschlossen werden, die mit folgendem Protokoll
 * an die serielle Schnittstelle ausgegeben werden:
 *
 * byte0  byte1      byte2      byte3      ...
 * A-H    0-9        0-9        0-9        n
 * Kanal  Millis                      Newline
 *
 * Die Millis sind dabei die Millisekunden zwischen den letzten beiden Low-High
 * auf dem angegebenen Kanal.
 */
const byte counterPins[8] = { 2,3,4,5,6,7,8,9 };
unsigned long millisBetween[8];
unsigned long lastMillis[8];
byte lastState[8];

void setup() {
  for (byte i = 0; i < sizeof(counterPins); i++) {
    pinMode(counterPins[i], INPUT);
    digitalWrite(counterPins[i], LOW);
    millisBetween[i] = 0;
    lastMillis[i] = 0;
    lastState[i] = 0;
  }
  Serial.begin(9600);
}
```

An den S0+ habe ich 5v anliegen, S0- geht über einen 4,5k gegen Masse und auf die Pin's. Das funktioniert.

Nun habe ich:

```
const byte counterPins[8] = { 2,3,4,5,6,7,8,9 };
```

in

```
const byte counterPins[1] = {30};
```

geändert- und den Rest dazu, das funktioniert auch.

Die Impulse werden zuverlässig empfangen. Die Länge des Impuls beträgt 30ms.

Bei 5v kommt da nicht mehr viel an, aber es wird zuverlässig erkannt.

Jetzt mein Problem.

Wie kann ich Pin 30 mit einer einfachen "if-Abfrage" abfragen:

```
If (pin30 == xx)
{
  mache was
}
```

Frage ich den Pin mit "if" ab, dann werden die Impulse nicht erkannt. Selbst wenn der Sketch nur aus der Abfrage besteht.

Warum funktioniert das mit dem Sketch, aber nicht mit einer "if-Abfrage"?

Ich kapier's nicht?

Wie kann ich das denn mit "if" erschlagen? Das muss doch möglich sein.

Gruß und Dank

Andreas

Quelle des Ganzen:

<https://blog.elektrowolle.de/2011/07/26/s0-messwerterfassung-von-stromzahlern/>

[HotSystems](#) 2 November 6, 2015, 10:30am

Das sollte so aussehen:

```
If (digitalRead(pin30) == xx)
{
  mache was
}
```

[Serenifly](#) 3 November 6, 2015, 10:33am

Wie kann ich Pin 30 mit einer einfachen "if-Abfrage" abfragen:

```
if(digitalRead(counterPins[0]) == HIGH)
```

Sowas? Oder mit dem Index einer for-Schleife. Wenn du nur einen Pin hast, kannst du die Arrays auch weglassen

Das ist übrigens überflüssig:

```
millisBetween[i] = 0;  
lastMillis[i] = 0;  
lastState[i] = 0;
```

Globale Variable werden automatisch mit 0 initialisiert

Außerdem kann man die globalen Variablen auch so machen dass sie automatisch skalieren:

```
const byte counterPins[] = { 2,3,4,5,6,7,8,9 };  
unsigned long millisBetween[sizeof(counterPins)];  
unsigned long lastMillis[sizeof(counterPins)];  
byte lastState[sizeof(counterPins)];
```

EDIT:

Die 8 in der letzten for-Schleife sollte man auch durch sizeof() ersetzen:

```
for (byte i = 0; i < 8; i++)
```

Dann werden nicht verwendete Kanäle nicht gesendet

[SkobyMobil](#) 4 November 9, 2015, 11:42pm

Hallo,

wie man die Pin's abfragt sollte ich eigentlich wissen- so wie Ihr es hier beschrieben habt.

Nachdem ich die Verdrahtung neu aufgebaut habe, funktioniert das auch. Weiß der Teufel, was dort vorher nicht gestimmt hat.

Die Verbesserungen von Serenifly sind ja nett gemeint, aber ich glaube nicht, das der Sketch es schafft jeden Impuls zu empfangen.

Wenn wir einmal von 230V ausgehen und 16A max. Last, dann sind wir bei 3680P/Wh.

Bei meinem Zähler wären das dann 7360 Impulse/Std, macht 2 Impulse/Sec. Das sind dann alle 500ms/Impuls.

Jetzt haben wir aber 8 Zähler, bleibt für einen Zähler eine Verarbeitungszeit von 63ms.

Der Sketch benötigt für einen Zähler 17ms. Bleibt ein Rest von 46ms- das muß dann reichen für Berechnung und Darstellung auf einem Display.

Läßt man nun noch eine Uhr für die Zeit und eine SD für den Zählerstand mitlaufen, dann wird die Kiste verdammt eng- glaube ich.

Gruß und Dank

Andreas

combie 5 November 10, 2015, 4:36am

dann wird die Kiste verdammt eng- glaube ich.

Dann kannst du deinen AVR noch reichlich in den Schlaf schicken.....

Da wird nix eng!

Es sei denn du baust blockierenden Code.

wie man die Pin's abfragt sollte ich eigentlich wissen- so wie Ihr es hier beschrieben habt.

```
if(digitalRead(counterPins[0]) == HIGH)
```

Ich verstehe nicht, warum man einen bool Wert mit HIGH vergleichen wollte...

Ein Beschäftigungsmaßnahme für arbeitslose AVRs?

```
if(digitalRead(counterPins[0]))
```

jurs 6 November 10, 2015, 12:16pm

SkobyMobil:

Jetzt mein Problem.

Wie kann ich Pin 30 mit einer einfachen "if-Abfrage" abfragen:

```
If (pin30 == xx)
{
  mache was
}
```

Frage ich den Pin mit "if" ab, dann werden die Impulse nicht erkannt.

Dein Problem scheint zu sein, dass Du bisher die einfache Schalterauswertung noch nicht verstanden hast, denn der S0-Ausgang ist nichts anderes als ein Schalter.

Und ob Du die Abfrage nun "Schalter wird gedrückt" oder "Impuls wird erkannt" nennst, spielt für die Software keinen Unterschied.

Erkennen mußt Du dabei nicht den aktuellen "Zustand" des Schalters, sondern Du mußt die "Zustandsänderung" erkennen, also wenn der Pegel wechselt.

In dem von Dir verlinkten Codebeispiel finde diese Erkennung der Zustandsänderung ("State change detection") hier statt:

```
byte val = digitalRead(counterPins[i]);  
if (val == HIGH && lastState[i] == LOW)  
{  
    // hier hat sich der Zustand geändert und der Impuls ist erkannt worden  
}
```

Und wenn Du die Zustandsänderung erkannt hast, mußt Du den letzten ausgelesenen Zustand in einer Variablen merken, das geschieht danach mit diesem Code:

```
lastState[i] = val;
```

[Serenifly](#) 7 November 10, 2015, 1:55pm

combie:

Ich verstehe nicht, warum man einen bool Wert mit HIGH vergleichen wollte...

Ein Beschäftigungsmaßnahmen für arbeitslose AVR's?

```
if(digitalRead(counterPins[0]))
```

Du hast noch nie Assembler programmiert, oder? Da mußt ein Vergleich irgendeiner Art stattfinden. Anders geht es nicht. Ob man da == HIGH/LOW, oder in anderen Fällen true/false schreibt hat mehr damit zu tun wie viel man tippen will, oder ob man es lesbarer findet.

Außerdem liefert digitalRead() einen int!

Mich wundert dass der Code nicht absolut identisch ist, aber der Unterschied ist minimal.

```
if (digitalRead(2) == HIGH)
{ ... }
```

wird zu:

```
ldi r24, 0x02
call 0x180 ; 0x180 <digitalRead>
sbiw r24, 0x01 ; 1
brne .+12 ; 0x12c <loop+0x16>
```

```
if (digitalRead(2))
{ ... }
```

wird zu:

```
ldi r24, 0x02
call 0x180 ; 0x180 <digitalRead>
or r24, r25
breq .+12 ; 0x12c <loop+0x16>
```

sbiw (subtract immediate from word) dauert 2 Zyklen, da der Befehl einen 16-Bit Operanden hat. Or dauert nur einen Zyklus.

Glückwunsch. Du hast einen Takt gespart.

[SkobyMobil](#) 8 November 10, 2015, 3:55pm

Hallo,
da haben mich die "richtigen" erwischt- jurs und Serenifly!

Bei euch muss man immer höllisch aufpassen was man macht, mit jedem Tip auf der Tastatur verschwindet die vertraute Sicherheit...

ich verstehe das schon- glaube ich...

Mit dem Beispiel-Sketch wurden Impulse erkannt.

Mit meiner einfachen Abfrage wurde kein Impuls erkannt-
mein Messgerät erkannte aber einen Impuls am Pin.

Es ging nur erst einmal darum, diesen Impuls zu erkennen- nicht darum, wie er später zu verwerten ist.

"S0-Ausgang ist nichts anderes als ein Schalter"

Im weitesten Sinne gebe ich Dir da recht, die ImpulsLänge beträgt 30ms.

Ich weiß noch nicht, wie genau es ist. Ich bin mir noch nicht recht im klaren darüber, ob die ImpulsLänge (30ms) mit in die Rechnung einbezogen werden muss.

Der S0-Impuls wird von einem geeichten Zähler übertragen. Nach knapp 3 Tagen zeigte dieser Zähler bei ca. 15,47P/Wh einen Stand von 930Wh an. Der Arduino einen von 924Wh. Das könnten im Jahr 730Wh sein.

Woher diese "Ungenauigkeit" kommt weiß ich noch nicht. Das könnten die millis() sein, Rundungsfehler oder eben eine falsche Impulsberechnung.

Gruß und Dank

Andeas

[jurs](#) 9 November 10, 2015, 4:55pm

SkobyMobil:

Nach knapp 3 Tagen

zeigte dieser Zähler bei ca. 15,47P/Wh einen Stand von 930Wh an.

Der Arduino einen von 924Wh. Das könnten im Jahr 730Wh sein.

Was bedeutet die Einheit "P/Wh", was ist das?

Und was ist das für ein Zähler?

Wenn das mal angenommen ein Zähler ist, der 1000 Impulse pro kWh liefert, also 1 Impuls pro Wh, dann ist der Unterschied zwischen 930 Wh und 924 Wh ein Unterschied von 6 Impulsen, die nicht gezählt wurden.

Die Frage ist: Was könnte in Deinem Code verhindern, dass diese 6 Impulse gezählt worden sind? Hast Du delay() Aufrufe in Deinem Programm drin?

Hast Du irgendwelche Aktionen, die länger als eine handvoll Millisekunden dauern, so dass ein 30 Millisekunden langer Impuls manchmal verpaßt werden könnte?

Das ist schwierig zu sagen, ohne den VOLLSTÄNDIGEN Code des Programms zu sehen.

Im übrigen ist Dein Programm von Dir ausdrücklich dafür vorgesehen, dass es um so fehlerhafter und ungenauer arbeitet, je größer die Zahlen werden. Und irgendwo zwischen 16 Mio. und 17 Mio. Wattstunden, also ca. zwischen 16000 und 17000 kWh, die Du mit Deinem Programm zählst, wird der Zähler komplett stehenbleiben und gar nicht mehr weiterlaufen. Der Grund liegt darin, dass Du Impulse, also ganze Zahlen, in float-Gleitkommavariablen zählst, die nur eine sehr begrenzte Auflösung und Genauigkeit haben.

[SkobyMobil](#) 10 November 10, 2015, 6:20pm

Hallo,
P/Wh, P ist die Leistung in Wh.
Der Zähler gibt per kWh 2000 Impulse raus.
Eltako-Zähler-Stand 930, Arduino-Zähler-Stand 1848, Mess-Zeitraum 59:10Std.

"dass Du Impulse, also ganze Zahlen, in float-Gleitkommavariablen zählst"

int 5 bleibt doch auch float 5.0 immer?
float 5.0*0.5 bleibt doch 2.5, nach .5 kommt doch nichts mehr?
Was ist da also ungenau, oder wie geht's genauer?

Ich nehme ja die gemessene Zeit auf.
116809ms bei 15.41Wh

Wenn jetzt ein Impuls nicht erfasst wird, dann müssten die 116809ms sich doch
ca. verdoppeln?

Wenn ich die 116808ms mit einer min/max Variable festhalte, dann könnte man
doch erkennen ob ein Impuls fehlt?

Jetzt habe ich z.B. 117744ms bei 15.29Wh

Ich könnte doch max auf 116808 legen, wenn max sich erhöht ist maxneu 117744
Wenn max > 200000, z.B.
dann
ZImp++

So könnte man doch einen fehlenden Impuls erwischen?
Ich habe Dir den ganzen Sketch einmal dran gehängt.
Gruß und Dank
Andreas

[jurs](#) 11 November 10, 2015, 7:05pm

SkobyMobil:
P/Wh, P ist die Leistung in Wh.

Eine Leistung P wird allerdings in Watt [W] gemessen, und nicht in [Wh] wie eine Arbeit.

SkobyMobil:
Der Zähler gibt per kWh 2000 Impulse raus.
Eltako-Zähler-Stand 930, Arduino-Zähler-Stand 1848, Mess-Zeitraum 59:10Std.

Machst Du da einen Vergleich desselben Zählers, indem Du

- einmal die Differenz des angezeigten Zählerstands Anfang/Ende ermittelst und
- einmal Impulse mit Arduino zählst und daraus die gezählte Arbeit ausrechnest?

Oder vergleichst Du zwei verschiedene Zähler miteinander?

[ElEspanol](#) 12 November 10, 2015, 7:55pm

Wieso ist überhaupt die Summe der Impulsdauern notwendig? Ich dachte, 2000 Impulse pro kWh, dann muss ich doch nur die steigenden Flanken zählen, um den Stromverbrauch zu haben. Und Impulse sind immer ganz und nie negativ, also unsigned long.

Aus dem Zeitabstand der steigenden Flanke kann man mehr oder weniger genau auf den momentan fließenden Strom schließen.

So zumindest habe ich das verstanden.

[SkobyMobil](#) 13 November 10, 2015, 8:19pm

Hallo,
den Speicher des Eltako stelle ich auf "0"
Dann starte ich den Arduino.

Der Eltako ist nach 59:10h auf 930 gesprungen.
Zur gleichen Zeit hatte der Arduino 1848 Impulse gezählt.
Bis zum nächsten Impuls wären es max 01:56min.
1848 Impulse sind (bei 2000Imp/kWh) 0,924kWh

Es besteht also ein Unterschied von 6W oder 12 Impulsen.
Die Wirkleistung hole ich mir über die gemessene Zeit zwischen 2 Impulsen.

Zeit 115530ms
 $(3600000/115530)*0.5=15.58W$

Für den Stand der Zähler (Eltako 0.930kWh zu Arduino 0.924kWh) werden nur die Impulse des Arduino gezählt.

Mit der Zeit, die zwischen 2 Impulsen vergeht, läßt sich die momentare Wirkleistung errechnen. Und die ist ziemlich genau. S.O.
Ich weiß nur noch nicht, ob die Impulslänge von 30ms mit in die Rechnung gehört. Es wird ja die Zeit zwischen 2 LOW-Flanken gemessen.

" Und Impulse sind immer ganz und nie negativ"

$2000\text{Imp/kWh} = 1\text{Imp}/0.5W$

Gruß und Dank
Andreas

[ElEspanol](#) 14 November 10, 2015, 8:48pm

was gibt denn der Hersteller für Genauigkeit an?

[SkobyMobil](#) 15 November 10, 2015, 8:58pm

Hallo,
Genauigkeitsklasse für $\pm 1\%$ B
S0 nach DIN EN62053-31
Gruß und Spaß
Andreas

[ElEspanol](#) 16 November 10, 2015, 9:11pm

Na ja, 12 zu 1848 sind ca. 0,6%
weil zwischen Impulsen und Display wird es ja auch diese Ungenauigkeiten geben

[mayrst](#) 17 December 2, 2015, 2:13am

Hi Leute, kann ich hier auch nochmal nachhaken?

ich habe mir nen Wechselstromzähler DRS155D von E-Tech besorgt, 1000 Imp/kWh

Habe ein Tutorial nachgemacht und den Code eingegeben.

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
#include <StopWatch.h>
```

```
StopWatch MySW;  
StopWatch SWarray[5];
```

```
void setup() {
```

```
pinMode(6, INPUT);
lcd.begin(16, 2);
lcd.print("LADEN...");
delay(500);
lcd.clear();
lcd.print("WARTE AUF IMPULS");
lcd.setCursor(0, 1);
lcd.print("<<<<<<<<>>>>>>>");
Serial.begin(9600);

SWarray[0].start();

}

void loop() {

if(digitalRead(6) == 0){

MySW.stop();
zeit = MySW.elapsed();
MySW.reset();
MySW.start();
imp = imp + 1;

delay(100);
while(digitalRead(6) == 0) {
delay(100);
}

kwh = imp / 1000;
lcd.clear();
lcd.print("KW");
dtostrf(kwh, 1, 3, tmp1);
lcd.setCursor(3, 0);
lcd.print(tmp1);

lcd.setCursor(0, 1);
lcd.print("EU");
lcd.setCursor(3, 1);
euro = kwh * Preis;
dtostrf(euro, 1, 4, tmp2);
Serial.print(tmp2);

lcd.setCursor(12, 0);
watt = 3600 / (zeit / 1000);

if (int(watt) > 9999){
lcd.print("OVER");
}
}
```

```
else {  
  lcd.print(int(watt));  
}  
lcd.setCursor(12, 1);  
lcd.print("WATT");  
  
}  
  
}
```

Nun hängt bei mir eine 60 Watt Lampe zum Test:

und beim Serial.print von kwh und watt erhalte ich

```
kwh 0.00  
watt inf  
kwh 0.00  
watt 318.22  
kwh 0.00  
watt 123.65  
kwh 0.00  
watt 216.67  
kwh 0.00  
watt 246.34  
kwh 0.01  
watt 192.36  
kwh 0.01  
watt 235.08
```

die kwh steigen dann ca. jede Minute um 0.02 und die watt eiern auch kurios herum 234, 453, ...

Bei ner 60 Watt Lampe sollte doch eigentlich bei kWh nach einer Stunde 0.06 stehen und 60 Watt bei der Liveanzeige.

Was mache ich den hier falsch?

[SkobyMobil](#) 18 December 2, 2015, 1:50pm

Hallo,
ich weiß nicht, was Du falsch gemacht hast, aber die "delay()" haben da nichts zu suchen...

Den Zähler richtig angeschlossen?

S0+ auf 5V

S0- über Widerstand gegen Massen, und weiter auf DigiPin

Irgendwie zu viel Gemurkstes, weiß nicht...

Kanst ja einmal probieren:

Das:

"PWh=(3600000/ZeitWatt)*0.5;"

müßtest Du vielleicht angleichen...

Gruß und Spaß

Andreas

P.S. das hier ist ein ArduinoForum- KEIN- FischertechnikForum

```
#include <LiquidCrystal.h>

const int SOA = 30; // S0-

//***** Für S0-Zähler *****
unsigned long StartMillis;
unsigned long MessMillis;
unsigned long EndeMillis;
unsigned long ImpulsZeit;
int val;
byte lastState;
float KWh;
float ZlKWh=-1;
float ZeitWatt;
float PWh;
unsigned long maxMessMillis;
static boolean SER = true;
//*****

LiquidCrystal lcd(12, 11, 5, 4, 7, 2);
void setup()
{
  pinMode(SOA, INPUT);
  digitalWrite(SOA, LOW);
}

void loop()
```

[jurs](#) 19 December 2, 2015, 2:09pm

mayrst:

Was mache ich den hier falsch?

Wenn bei Dir ein Programm läuft und etwas falsches anzeigt, dann bestimmt nicht das von Dir gepostete Programm. Das läßt sich ohne eine Deklaration etlicher Variablen, die darin verwendet werden, nämlich nicht einmal kompilieren.

Außerdem würde ich gerne mal das genaue Schaltbild sehen, wie Du Arduino und S0-Schnittstelle verbunden hast und den PullUp-Widerstand verwendest. Wenn Du keinen PullUp-Widerstand in der

Schaltung hast, würdest Du im Programm auch den falschen pinMode setzen und das Programm kann alleine deshalb nicht korrekt funktionieren, weil der Code nicht zur angeschlossenen Hardware passt.

Abgesehen davon programmierst Du da zum Zählen von Zeiten und Impulsen einen völligen Irrsinn:

```
if(digitalRead(6) ==0){  
  
    MySw.stop();  
    zeit = MySw.elapsed();  
    MySw.reset();  
    MySw.start();  
    imp = imp + 1;  
  
    delay(100);  
    while(digitalRead(6) == 0) {  
        delay(100);  
    }  
}
```

Du zählst Impulse, indem Du eine Stoppuhr anhältst und die Zeit abliest, dann die Stoppuhr resettest und wieder startest? Und dann wird mit delay() das laufende Programm noch mindestens einmal, eventuell aber auch mehrmals für eine Zehntelsekunde blockiert, bevor es weiterläuft? Bist Du sicher, dass die mit der Stoppuhr gemessene Zeit dabei noch irgendwas mit der Zeit zwischen zwei Impulsen zu tun hat?

Mein Tipp: Schau Dir mal das Arduino-Programmierbeispiel "StateChangeDetection" an, und verwende statt Deiner Stoppuhr-Library mit der wild gestoppten, abgelesenen, resetteten und wieder gestarteten Stoppuhr den Stand des "millis()" Zeitzählers zur Ermittlung der Zeiten!

Also mache ein vernünftiges Programm statt so eines Kuriosums!

Und prüfe, ob Deine Hardwareschaltung (PullUp-Widerstand) zum pinMode im Programm passt!

[mayrst](#) 20 December 3, 2015, 10:06pm

Hi SkobyMobil

ich habe es mal mit deinem Script versucht, leider auch ohne Erfolg,

sollte es oben 3 und nicht 30 heißen? hast du dich da vertippt?

Als input wähle ich also Digital 3 als Input?

Zum Thema wie schließe ich das ganze an:

S0+ auf Arduino 5V

S0- über 4,7 kOhm auf Pin D3

habe auf folgender Seite gelesen dass der Arduino interne Widerstände zum schalten hat und dass man den Analog In auch nutzen kann.

<http://www.msxfaq.de/verschiedenes/bastelbude/arduin0s0.htm>

[jurs](#) 21 December 3, 2015, 10:32pm

mayrst:

Zum Thema wie schlieÙe ich das ganze an:

S0+ auf Arduino 5V

S0- über 4,7 kOhm auf Pin D3

Hört sich schon mal falsch an.

Richtig beschrieben wäre die korrekte Schaltung wohl eher so:

- S0+ auf Arduino 5V
- S0- auf Pin D3
- Zusätzlich Pin D3 über Widerstand 4.7K an GND

[SkobyMobil](#) 22 December 3, 2015, 10:46pm

Hallo,

der Sketch von mit läuft auf einem Mega 2560 fehlerfrei (daher Pin 30)

Er gibt über Serial die ms zwischen 2 Impulsen raus.

Rechnet hier ($P_{Wh} = (3600000 / \text{ZeitWatt}) * 0.5$;) P/Wh raus- für einen Zähler, der 2000Imp/kWh macht.

Du kannst lesen was Du willst, verdrahte so, wie jurs es beschrieben hat.

Gruß und Spaß

Andreas

[mayrst](#) 23 December 4, 2015, 1:41am

OK Sry wegen Fischertechnik, aber für manche ist einfach e alles klar andere müssen sich halt erst einarbeiten.

jetzt klappt mein Skript, hab komplett gewechselt.

```
//Anzahl der Pulse für die Leistungsmessung
```

```
long pulseCount = 0;
```

```
//Werte für gemessene Leistung
```

```
unsigned long pulseTime,lastTime;
```

```
//Leistung und Energie
```

```
double power, kWh;
```

```
//Anzahl der Pulse pro Wh (auf Stromzähler)
```

```
int w_per_pulse = 1; //1000 pulses/kwh = 1 pulse per wh
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
// bestimmt Digital Pin3 als Messeingang
```

```
attachInterrupt(1, onPulse, RISING);
```

```
}
```

```
void loop()
```

```
{
```

```
}
```

```
void onPulse()
```

```
{
```

```
//Zeit zwischen den Pulsen
```

```
lastTime = pulseTime;
```

```
pulseTime = micros();
```

```
//pulseCounter
```

```
pulseCount++;
```

```
//Ermittlung der Leistung
```

```
power = (3600000000.0 / (pulseTime - lastTime))*w_per_pulse;
```

```
//Ermittlung der kWh
```

```
kWh = (1.0*pulseCount*w_per_pulse)/1000; //multiply by 1000 to pulses per wh to kwh convert wh to kwh
```

```
//Ausgabe
```

```
Serial.print("Power");
```

```
Serial.print(power);
```

```
Serial.print(" ");
```

```
Serial.print("kWh:");
```

```
Serial.println(kWh);
```

```
}
```

[mayrst](#) 24 December 4, 2015, 1:57am

Dürfte ich nochmal um eine Idee fragen!

Wenn ich nun meine Last ein- und ausstecke werden 3 undefinierte Werte gemessen, kann man dies verhindern?

Idee: beim messen vom ersten Puls 20 Sekunden sleep

was meint ihr? Habt ihr n besseren Vorschlag?

[jurs](#) 25 December 4, 2015, 2:54pm

mayrst:

was meint ihr? Habt ihr n besseren Vorschlag?

Also ich habe zwei Vorschläge:

1. Vorschlag

Wenn Du von Interruptprogrammierung sowas von "keine Ahnung" hast, wie Du es hier mit dem geposteten Code zeigst, dann programmiere das Auslesen der Impulse OHNE INTERRUPTS ZU VERWENDEN! Das ist sowohl einfacher als auch zuverlässiger. Insbesondere für Anfänger ohne Ahnung von Interruptprogrammierung. Laut Wikipedia-Eintrag zur [S0 Schnittstelle](#) dauert ein S0-Impuls 30 Millisekunden, und um so langsame Impulse zu erkennen ($30\text{ms} = 30000\mu\text{s}$), braucht es keine Interrupts. Jedenfalls nicht, wenn Du ein Programm auf die Reihe bekommst, das Dein Programm nicht mit "delay()" absichtlich für längere Zeit blockiert.

2. Vorschlag

Wenn es denn unbedingt ein Programm mit Verwendung von Interrupts sein muss, dann informiere Dich wenigstens über ein paar Grundlagen der Programmierung mit Interrupts, z.B. wie RAM-Speicher und Variablen konsistent gehalten werden, wenn zwei Threads gemeinsam auf dieselben RAM-Bereiche zugreifen ("Shared Memory" zwischen normalem Code und Interrupt-Code)! Profi-Programmiertechniken sind eher für Programmierer geeignet, die wenigstens ein paar grundlegende Anfänger-Grundlagen bereits beherrschen.

mayrst:

Idee: beim messen vom ersten Puls 20 Sekunden sleep

Ein Programm, das auch nur ein einziges mal "delay(irgendwas)" verwendet, hat bereits mit diesem einen Funktionsaufruf als Mikrocontroller-Programm total verkackt, und Du kannst Dir stattdessen ein neues Programmkonzept überlegen, das kein "delay()" verwendet. Und wenn "delay()", dann sollte die Zeit deutlich kürzer sein als die kürzeste Zeit, mit der Dein Programm umgehen muss. Diese kürzeste Zeit scheint in Deinem Programm die Impulsdauer eines S0-Impulses zu sein, ca. 30 ms. Also sollte Dein Programm pro einmaligem Durchlauf durch die loop() Funktion aufsummiert auf keinen Fall mehr als delay(15) verwenden, um die Impulse mit Sicherheit zu erkennen, ohne dafür Interrupts zu benötigen.

[SkobyMobil](#) 26 December 4, 2015, 9:06pm

Hallo,

"Wenn ich nun meine Last ein- und ausstecke werden 3 undefinierte Werte gemessen, kann man dies verhindern?"

Das kann man (auch Frau)...

Erst einmal in der BDA des WSZ lesen, was der überhaupt kann und von welcher (zweifelhaften) Qualität der denn ist.

Dann das ganze Zeug's richtig und sicher verdrahten.

Wichtig wäre auch einen Sketch zu verwenden, den der geübte Laie auch versteht.

Du kannst ja mal etwas probieren.

Mit dem S0 ist es ja möglich, den tatsächlichen Verbrauch zu messen, so wie auch die momentare Wirkleistung.

Du nimmst zwei Glühbirnen, eine 100W und eine 60W. Die 100W läßt Du 5Min leuchten, dann umschalten auf die 60W.

Die gemessene Wirkleistung wird Dir ja sofort angezeigt, den genauen Verbrauch kannst Du Dir errechnen und mit dem Arduino vergleichen.

Auch ein Wasserkocher mit $\geq 2\text{Kw}$ oder ein Fön mit $1,8\text{Kw}$ können sehr hilfreich sein.

Dann würde ich einmal über mehrere Tage messen, um zu schauen wie "genau" das ganze denn so sein könnte.

Bei mir habe ich z.B. einen Unterschied von 6W in drei Tagen. das sind 60W im Monat oder 720W im Jahr. Das ist fast ne´ kWh.

Du bemerkst also, das man so eine Geschichte ziemlich sicher und "genau" aufbauen kann.

Zu Störungen bei plötzlichen Lastwechseln, gleich welcher Art, darf es auf keinen Fall kommen. Egal was Du im laufendem Betrieb an Deinem WSZ anschließt- oder abstöpsels, weder der Arduino- noch der WSZ dürfen dabei einen Muks von sich geben.

"Idee: beim messen vom ersten Puls 20 Sekunden sleep"

"sleep" hört sich für mich wie delay() an, geht überhaupt nicht.

Warum nimmst Du nicht die Zeit mit dem zweiten Impuls?

Beim ersten Impuls "StopUhr" an, beim zweiten "aus".

Gruß und Spaß

Andreas

P.S. passe auf, das Du keine gewischt kriegst- geht ganz schnell bei KamikazeVerdrahtungen

Related topics

Topic	Replies	Activity
<u>S0 Impulse für 10 Sekunden zählen</u>	3	May 6, 2021
<u>Einstellbaren Impuls ausgeben</u>	8	May 6, 2021
<u>Fragen zu attachInterrupt +S0 Schnittstelle auslesen</u>	23	January 22, 2025
<u>Flaschenhals Serielle Schnittstelle oder schlechter Code ?</u>	13	May 6, 2021
<u>S0-WSZ simulieren</u>	7	May 6, 2021