LINKS

http://www.electronics-lab.com/project/pir-sensor/

https://learn.adafruit.com/memories-of-an-arduino/optimizing-sram

https://hackingmajenkoblog.wordpress.com/2016/02/04/the-evils-of-arduino-strings/

https://miscsolutions.wordpress.com/2011/10/16/five-things-i-never-use-in-arduino-projects/

https://www.itead.cc/wiki/Arduino\_GPS\_shield

\*\*\*PS2

http://www.computer-engineering.org/ps2protocol/

https://github.com/kristopher/PS2-Mouse-Arduino

https://github.com/lavz24/Flick\_Arduino/tree/master/libraries/ps2

http://code.rancidbacon.com/ProjectLogArduinoUSB

http://dkudrow.blogspot.com.br/2013/08/ps2-keyboard-emulation-with-arduino-uno.html

https://www.win.tue.nl/~aeb/linux/kbd/scancodes-13.html

http://www.eecg.toronto.edu/~jayar/ece241\_08F/AudioVideoCores/ps2/ps2.html

http://www.eecg.toronto.edu/~jayar/ece241\_08F/AudioVideoCores/ps2/ps2.html#apkeycodes

http://retired.beyondlogic.org/keyboard/keybrd.htm

http://wiki.osdev.org/Mouse\_Input

http://maxembedded.com/2013/06/using-an-analog-accelerometer/

http://bildr.org/2011/04/sensing-orientation-with-the-adxl335-arduino/

http://bildr.org/2011/03/adxl345-arduino/

https://chionophilous.wordpress.com/2011/06/22/accelerometer-calibration-i-introduction/

http://www.hownottoengineer.com/projects/rgb-led-cube.html

\* Faltou o sensor piezoelétrico

\* Faltou um sensor de infravermelho da sharp com saída analógica e não digital.

\* Faltou pelo menos dois encoders.

\* Faltou um magnetômetro.

\* Faltou um controle de Wii

\* Faltou um controle de playstation2

######################################

PROJETOS A FAZER

- arduino com acelerometro agir como as setas de um teclado ps2.

- controlar o carrinho com um controle de acelerometro.

$$$$$$$$$$$$

Esta parte do código tem a ver com interrupções;

uint8\_t oldSREG = SREG; //salva o status do registrador de interrupções

cli(); // para as interrupções

Pos = encoderPos; // faz o que o código pede

SREG = oldSREG; // restaura o registrador de interrupções e sua funcionalidade.

LDR --> é possível obter os valores de luminosidade do ambiente com um LDR. Fiz um sketch que retorna o valor que está sendo lido para a serial

um em que é possível colocar o limite e estipular o dia e a noite.

Este é o esqueleto principal de um sketch arduíno.

int main(void)

{

init();

setup();

for (;;)

loop();

return 0;

}

TYPES

Interessante sobre strings:

Além de utilizar o objeto String que tem mais funcionalidades a um custo maior de memória, é possível declarar strings de várias outras formas.

char Str1[15];

char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};

char Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'};

char Str4[ ] = "arduino";

char Str5[8] = "arduino";

char Str6[15] = "arduino";

O compilador insere automaticamente o caractere nulo '\0'.

Quando uma string for instanciada dentro de "" o compilador ajusta o tamanho da variável na memória e adiciona o caractere nulo no final da cadeia de chars.

Se uma string for declarada com tamanho definido e não incluir o espaço para o caractere nulo e nem declarar o caractere nulo, pode ocorrer problemas na execução do programa.

Long strings:

char myString[] = "This is the first line"

" this is the second line"

" etcetera";

Um array de strings necessita ser declarado como um array de ponteiros char:

char\* myStrings[]={"This is string 1", "This is string 2", "This is string 3",

"This is string 4", "This is string 5","This is string 6"};

char string [] = "Arduino";

strlen(string); --> devolve o tamanho da string;

strcpy(destination,source) --> copia uma string na outra;

strncpy(destination,source,n)--> limita o numero de caracteres a copiar;

strcat(destination,source) --> concatena as strings;

strcmp(str1,str2) --> compara strings;

Este código pega os valores dos sensores e transformam em porcentagem.

int sensorPercent(int pin)

{

int percent;

}

int val = analogRead(pin); // read the sensor (ranges from 0 to 1023)

percent = map(val,0,1023,0,100); // percent will range from 0 to 100.

return percent;

Debounce Button

O sketch do livro é bom que é possível utilizar vários botões por ser uma função. Contudo para realizar um toggle, o sketch do site do arduino é bem mais consistente. Só que não consegui colocar em uma função. Existe a classe Bounce2 que também não achei consistente para fazer um toggle. Contudo, o debounce é para retirar aquela variação inicial do botão. Eu é que quero fazer um toggle.

no livro mostra um detalhe interessante de como utilizar um hold button e ir incrementando o valor exponencialmente. sketch buttonHoldTime.

\* KEYPAD

Bibliotecas novas até agora:

- Bounce2;

- Keypad.

Trabalhando com keypads é muito interessante. Tenho aquele keypad flat que possui 16 posições. um teclado hexanumérico.

Possui 8 pinos de saída. os 4 primeiros são as linhas e os outros 4 são as colunas.

A ideia é que 4 sejam input e os outros output. No arduíno é possível colocar como pull-up, e a tecla acionada estará em down. Ou fazer pull-up e pull-down em hardware.

Provavelmente é o estilo de um cubo de led. Como se fosse uma camada.

No caso para saber se uma tecla foi pressionada, é feito um rastreio das teclas, pelo jeito quanto maior o numero de teclas mais lento será.

Fiz um teste e o tempo que demorou para varrer o teclado 4x4 foi de 140 a 144 microsegundos. bem rápido.

Utilizar a biblioteca Keypad torna o uso muito simples. E os códigos de exemplo são muuuuito interessantes. Multikey, eventos... muito completo e complexo.

POTENCIÔMETROS.

Ao utilizar potenciômetros, percebi que a variação de resistência, ou os valores lidos na porta analógica, vai aumentando de acordo com a posição do potenciômetro. quer dizer, no início a variação é pouca, mas para perto do fim ela é maior, e os valores mudam mais rapidamente.

LEITURA ANALÓGICA.

utilizei o Chip CD4051BE Multiplex Analógico para expandir as portas analógicas e funcionou. Fiz a leitura com nenhum delay entre as leituras e não ocorreu erro. leu perfeitamente.

O algorítimo para criar a tabela verdade foi interessante.

ABC S

000 0

001 1

010 2

011 3

100 4

101 5

110 6

111 7

A função que faz a leitura, recebe um número referente ao canal que vai ser lido. Para gerar os bits correspondentes foi feito o seguinte algoritimo

um for varre os bits 0 1 2. Para cada bit ele lê o bit correspondente com a função bitRead.

e escreve na porta correspondente a entrada da tabela verdade.

VOLTAGEM DE BATERIA EM UMA ANALOG INPUT

essa é uma abordagem importante, saber quanto de voltagem ainda tem uma bateria. As entradas analógicas medem tensão e dependendo da placa a tensão de referência é 5V ou 3.3V.

Para medir a voltagem da bateria com o arduino, basta ligar a bateria, o negativo no ground e o positivo na entrada analogica. Contudo, somente voltagens até a voltagem de referência da board.

Medindo a voltagem utilizando variáveis float, ocupam mais memoria do que com inteiros. Para eficiência, uma jogada seria pegar o valor em milivolts.

Como:

cada passo é 5V/1023... então para ter uma boa resolução você aumenta o valor da referência. No caso para ficar bem parecido com a leitura com float, usa-se o passo 500000/1023.

depois desse passo, multiplica pelo valor obtido na porta analogica. e por fim para mostrar em milivolts, basta dividir pelos números de zero a mais, que no caso são dois de 500000.

A resposta será então em milivolts, sem perder a resolução.

Para ler voltagens maiores que 5V será necessário utilizar um divisor de tensão. No caso de 9V, pode-se dividir pela metade, isto é 4,5V.

Fiz os divisores de tensão com 3 valores diferentes de par de resistores e constatei que os mesmos leram a mesma voltagem. O beneficio foi que quanto maior o valor da resistencia, menor foi a corrente.

1K, 4K7 e 10k.

SENSORES:

Sensores provêem informações das seguintes formas:

- digitais;

- analógicas;

- pulsos;

- serial;

- protocolos sincronos - I2C e SPI

SENSOR ULTRASÔNICO.

Tive alguns problemas com o meu sensor.

MODELO: HC-SR04 ele faz uma leitura correta de logo após uma leitura comoo 0, como se não tivesse nada na frente, então para saber se não existe nada na frente realmente é preciso fazer trÊs medições e se as três forem 0, significa que está fora do alcance do sensor.

Outro ponto que vi como um problema é que quando conecto outros dispositivos utilizando o mesmo vcc ele também não lê legal. Coloquei um PIR e um servo e ocorreram interferências nas leituras.

Pelo jeito a biblioteca Newping resolveu o problema da leitura.

Usando os sensores de infravermelho para obstáculos, notei que ele somente enxerga por uns 7cm. Ele não reflete a cor preta. O sensor que tenho envia um sinal digital e não analógico.

SENSOR DE TEMPERATURA LM35

Esse sensor de temperatura provoca uma alteração de 1mV por 0,1oC, isto é 10 milivolts por grau.

Sua precisão é de 0,5oC.

LEITOR DE RFID DE 125KHZ

Informações importantes no link: http://www.automalabs.com.br/modulo-rfid-125khz-com-saida-uart-rdm6300/

Agora os testes são com o leitor de rfid de 125khz. O modelo que tenho em mãos é o RDM6300. As conexões dele são simples.

Não fiz muitas pesquisas até o momento, primeiro quero ver seu funcionamento. O que sei até agora é que sua alimentação é de 5V e ele se comunica com o arduíno através da serial.

de um lado ele tem uma sequencia de 5 pinos. O primeiro pino é seu tx, que se comunica ao rx do arduíno. Nos testes estou utilizando SoftwareSerial nas portas 2,3 rx,tx.

Do outro lado ele tem um grupo de 3 pinos. Pino 2 VCC (pino do meio) e o 3 pino é o gnd.

Depois do mesmo lado tem mais dois pinos juntos. Estes pinos se ligam à bobina do leitor rfid, não importando a ordem.

Fiz um teste imprimindo tudo que vem pela serial em vários formatos. Esta foi a saída.

id: BIN, HEX, DEC, CHAR

0 10 2 2 #

1 110000 30 48 0

2 110000 30 48 0

3 110000 30 48 0

4 110000 30 48 0

5 110000 30 48 0

6 110000 30 48 0

7 110001 31 49 1

8 110100 34 52 4

9 110101 35 53 5

10 110000 30 48 0

11 110100 34 52 4

12 110100 34 52 4

13 11 3 3 #

O que dá pra notar é que são 14 bytes. O primeiro byte é o 2 decimal ou 10 binário. Na tabela ASCii representa STX. depois vem uma sequência de 12 bytes contendo a serial do cartão. Para finalizar o código vem o valor 3 decimal ou 11 binário. Na tabela ASCii representa ETX.

Se deixar o o parado na frente do leitor ele lê o código continuamente. O ideal seria criar estágios com um tempo para leitura. Um tempo que fosse necessário para o usuário colocar e tirar o cartão.

Um sketch legal foi verificar se o byte lido era o início de uma tag. depois disso, gravar a tag em um buffer de char. quando chegar o byte final, mostrar o valor da tag. o próximo passo será transformar a tag em um número do tipo long.

Descobri o protocolo da tag rfid de 125khz

a seguencia de 12 bytes fica da seguinte forma. 10 bytes representam os digitos hexadecimais da tag e os dois ultimos bytes representam os digitos hexadecimais do checksum.

Isto é, para calcular o checksum da tag, você precisa pegar de dois em dois caracteres e transformá-los em um número hexadecimal. Fazer um xor com todos os números. O resulta do é transformado em dois digitos hexadecimais de texto, isto é char.

Contudo, acredito que a verificação é desnecessária. Pois para autorizar as tags é preciso conhecê-las antes.

Criei uma classe!!!!!

\*\*\*\*\*\*\*\*\*\*\*

6.10. Não possuo um rotary encoder no momento, então não tem como testar.

Estes encoders são interessantes e sei que é utilizado em carrinhos e em máquinas com motores dc que não são de passo.

Eles podem ter resoluções diferentes. Enviam sinais de high e low pelos seus terminais.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PS2/MOUSE

Consegui um conector fêmea ps2 e adaptei em uma protoboard universal para conectar o mouse. Baixei a biblioteca ps2. nela tive que modificar o #include "WProgram.h" para #include "Arduino.h". Para funcionar e não gerar erro, tive que fechar e abrir o Arduino IDE sem compilar.

O difícil no caso é fazer com que o arduíno se comporte como um dispositivo ps2. vou pesquisar se é possível.

Encontrei uma outra biblioteca chamada PS2Mouse que faz o mesmo esquema que a ps2. Contudo, parece que o código é mais robusto. vou verificar os fontes.

Encontrei uma bibliioteca mais nova e remasterizada da ps2 que possui a função do scroll.

PS2DEV emula um dispositivo ps2. Consegui pegar um exemplo e rodar como teclado e rodou mais ou menos. como mouse não rodou o exemplo.

As emulações funcionam mais ou menos como teclado, como mouse não obtive sucesso em nenhuma tentativa. Vou deixar de lado por enquanto para não atrapalhar meus estudos, mas vou tentar melhorar os códigos para utilizar um teclado ps2.

Encontrei algumas bibliotecas para utilizar um teclado ps2 com o arduino. Fiz somente alguns testes e vi que dá para fazer algumas modificações, mas não queria me prender a isso no momento. Fiz somente os testes das bibliotecas.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GPS MODULE.

Tem vários detalhes que quero escrever aqui sobre este assunto. Contudo, não quero deixar de registrar alguns testes simples.

Antes de procurar sobre especificações, fiz um teste básico com o arduino e o shield GPS da Itead.

O shield é bem simples e fácil de conectar. Ainda não possuo uma antena, com isso, pode ser que o sinal não pegue em alguns lugares. Mas parece que em casa, sem antena, não teve problemas.

O shield possui dois jumpers que conectam os pinos tx e rx do shield. Você pode remov⁻lo do padrão e conectar em outros pinos. Estes outros pinos representam os pinos D0 até D7 do arduino. Com isso, não conectei na serial padrão. Fiz um sketch simples com o softwareSerial com RX/TX nos pinos 6/7 modifiquei o jumper do shield GPS e utilizei a serial como DEBUG.

A velocidade de conexão da UART do shield é de 9600, mesmo e bastou ligá-lo, subir o sketch, que os dados começaram a aparecer na serial.

19/09 - Na verdade sem antena não funcionou em casa. Comprei uma antena com 3 metros de fio e funcionou. Contudo, verifiquei que a antena pequena que veio no shield de gprs também funcionou..

A biblioteca tinyGPS é ótima e bem completa. Contudo, não vejo aplicações solitárias no momento. quero integrá-las com outras coisas. Vou deixar mais para frente também. Uma ideia será integrar os dados do GPS com python e pygame.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

####### ACELERÔMETRO.

O acelerômetro que tenho é o ADXL335 com saídas analógicas. e tenho o MPU6050 que é um acelerômetro e um giroscópio.

22/09/16

Estou fazendo alguns testes com o ADXL335, encontrei algumas libraries. Estou testando a biblioteca ADXL335, ela retorna phi, rho e theta. Contudo não estou entendendo o retorno dos valores.

Lendo o site http://maxembedded.com/2013/06/using-an-analog-accelerometer/, verifiquei o porquê a variação do acelerômetro não é alta. Pelo artigo os eixos x,y variam de 1,35V até 1,65V aproximadamente e o eixo z de 1,2V até 1,8V. E em descanço 1,5V em zero g bias.

Um detalhe interessante foi colocar o pino ARef do Arduino em 3,3V e no código incluir o comando

analogReference(EXTERNAL). Com isso a leitura ficou melhor.

#################################

VISUAL OUTPUT

\* \* Procurar pelo CI ULN2003A que parece controlar várias entradas e saídas com até 500mA.

DADO INTERESSANTE

\* \* Se for usar a função analogWrite, não é necessário configurar as portas como OUTPUT.

Na parte de RGB tem um algorítimo interessante de hueToRGB, tipo fazendo a roda de cores, primárias e secundárias.

Muito bacana este algorítimo de hueToRBG, ele faz passar pela roda de cores. Preciso entender melhor o processo para poder selecionar uma cor dentre todas.

22/05/2017

Piezos.

Vi que o piezo trabalha quase igual ao microfone. Só que tem que tocá-lo para produzir a tensão nas extremidades.

Estou sem o resistor de 1M então utilizei um de 4M7.

Na serial ele estabiliza entre 50 e 60. E quando toco nele ele vai para um valor alto e cai a zero.

Fazendo alguns ajustes de numeros de samples para leitura e threshold foi possível ascender o led de maneira ideal. Contudo é algo que precisa ser bem parametrizado.

20/11/2017

CONSEGUI REALIZAR UMA FUNCAO QUE AJUSTA O FPS DE EXECUCAO DO SKETCH

void tick(){

if (millis() - time\_passed > 1000){

if (fps > fps\_max){

delay\_ms = int(fps / ((fps / 1000) \* fps\_max));

}

Serial.println(fps);

Serial.println(delay\_ms);

time\_passed = millis();

fps = 0;

}

fps++;

Esta funcao é chamada dentro de um laco, ele faz a contagem de execucao deste laco. Cada execucao é guardada dentro da variavel fps. A cada segundo é verificado se o valor da variavel fps é maior que o valor maximo estipulado pelo usuario. Caso seja ele faz o calculo do valor que deve ser colocado dentro de um delay para que o fps\_maximo seja respeitado.

Calculo:

pega-se o numero de execucoes. Divide este numero pela sua milesima parte, que corresponde a quantidade de vezes por segundos, multiplica pelo maximo fps. Assim dará um valor que precisa ser colocado no delay para que o fps corrente diminua até atingir o valor do fps maximo.

Loko.