

Visão Estereoscópica no Arduino

Alves Junior, R. P.

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
Rildojunior6@hotmail.com

Feitosa, M. A. F.

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
marcilio@poli.br

Esteves, G. R. P.

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
gustavorpe@gmail.com

Resumo

Este trabalho apresenta a tentativa de utilizar o Arduino, uma plataforma de prototipagem eletrônica, no processo de visão estereoscópica, com a função de comandar uma câmera que capture duas imagens distintas do mesmo objeto. Após, o Arduino irá processar essas imagens para encontrar um ponto em comum e calcular a distância do objeto para as câmeras. Serão apresentados testes com sensores de mouse óptico, e com um módulo de câmera serial, mostrando a possibilidade deles se comunicarem com o Arduino, mas ao mesmo tempo justificando a dificuldade de atuarem em visão estereoscópica. O Galileo é citado como o Arduino ideal para trabalhar com estereoscopia devido ao seu poder de processamento. Porém será explicado a não realização dos testes com o mesmo, que foram impossibilitados por problemas com sua IDE particular.

Abstract

This article presents the attempt of using the Arduino, an electronic prototyping platform, in the stereoscopic vision process, to control a camera that captures two distinct images of the same object. After, the Arduino will process these images to get a point in common and calculate the distance of the object to the camera. Tests with optic mouse sensors and with a serial camera module will be presented, showing the possibility of them to communicate with the Arduino, but will be explained at the same time the difficulty of these instruments to work in stereoscopic vision. Galileo is mentioned as the ideal Arduino to work with stereoscopy because of its processing power. However, will be explained that the tests didn't happened with it, because of the problems in its particular IDE.

1 Introdução

A estereoscopia visual é o processo pelo qual a partir de duas imagens distintas, mas que contemplem a mesma região, pode se ter uma noção da tridimensionalidade dessa região. Este processo pode ser empregado para solucionar diversos problemas como o mapeamento de ambientes, cálculos de distâncias e reconhecimentos de objetos tridimensionais [1].

As práticas mais comuns em visão estereoscópica utilizam um ambiente computacional para a implementação da visão, realizando os cálculos necessários. Tal ambiente permite um melhor desempenho no trabalho com imagem resultando em maior rapidez no processamento da mesma.

Com o advento e a disseminação que está acontecendo com o Arduino, tentou-se nesse projeto analisar a possibilidade de se trabalhar visão estereoscópica nessa plataforma.

2 Plataforma Arduino

Arduino é uma plataforma de prototipagem eletrônica criada com o objetivo de permitir o desenvolvimento de controle de sistemas interativos, de baixo custo e acessível a todos. Ele faz parte do conceito de hardware e software livre e está aberto para uso e contribuição de toda sociedade [2].

Um grande diferencial deste dispositivo é ser mantido por uma comunidade que trabalha na filosofia *open-source*, desenvolvendo e divulgando gratuitamente seus projetos. [3]

Existem vários modelos de Arduino [4] com características de processamento parecidas, dependendo do microcontrolador utilizado pela placa. Na tabela 1 temos, respectivamente, qual o tipo de microcontrolador, de microprocessador, de memória flash, SRAM, EEPROM e frequência de *clock*, de alguns modelos.

Tabela 1. Comparativo de placas Arduino.

Arduino UNO	Duemilanove	Galileo
ATmega328	ATmega168/328	-
-	-	Intel Quark
32kB	16kB/32kB	8MB
2kB	1kB/2kB	512KB
1kB	512bytes/1kB	11KB
16MHz	16MHz	400MHz

3 Sensor de Mouse Óptico

O mouse ótico foi projetado no final da década de 90. Atualmente pode ser construído com uma câmera que processa mais de 1500 imagens por segundo. Possui um processador que digitaliza essas imagens a partir da comparação com um determinado padrão e as imagens captadas pelo sensor são digitalizadas e processadas [5].

O sensor CMOS (*Complementary Metal-oxide-semiconductor*) usa uma tecnologia de navegação óptica, em que é possível medir a mudança de posição através da aquisição de uma sequência de imagens da superfície. A partir disso, é determinada matematicamente a direção e a amplitude do movimento [6].

Viu-se a possibilidade de usar o sensor de mouse óptico como uma câmera, comandada pelo Arduino, e dessa forma obter as posições x e y de uma imagem, mesmo de baixíssima qualidade e resolução. Mesmo se configurando uma forma bem simples de capturar imagem, foi realizada tal experiência com a intenção de trabalhar com esses pontos, e realizar o cálculo da distância do sensor para a imagem obtida.

Porém um fato preocupante é o alcance óptico do mouse, que não ultrapassa 1,25 mm [5].

4 Módulo de câmera serial TTL

Para utilizar uma câmera no processo estereoscópico com o Arduino, foram investigadas as que pudessem estabelecer uma comunicação com a placa, já que o Arduino não possui entrada USB, a típica interface das câmeras mais comuns, as *webcams*.

Tentou-se utilizar duas câmeras para o projeto, por isso a câmera VC0706 [7] foi escolhida. Em sua comunicação com o Arduino ela utiliza apenas quatro pinos dos seis que possui, e como o arduino mais comum, o uno, possui poucos pinos de entrada e saída, fica mais viável utilizar essa câmera.

A câmera possui como principais características: sensor CMOS 0.3M pixels; formato de saída da imagem em JPEG (*Joint Photographic Experts Group*); Velocidade de 30fps (frames per second); distância de monitoração máxima de 10 metros; resolução máxima de 640x480 pixels; taxa de transmissão padrão de 38400 b/s; tensão de operação +5V, e de comunicação 3.3V.

A VC0706 é mais indicada para monitoração remota, segurança e detecção de movimento, segundo o fabricante [7].

A comunicação com a câmera é feita por sua interface serial [7]. A mesma desempenha diversas funções, e cada função tem seu comando específico.

4.1 Cartão de memória

Para armazenamento das imagens capturadas pela câmera, é utilizado em um cartão de memória. Para isso utilizamos um módulo cartão SD que, junto com o micro-SD e o cartão de memória, se comunicam com o Arduino por seis pinos dos oito que o módulo possui: pino CS (*Chip select*); pino MOSI (*Master Out Slave In*); pino MISO (*Master In Slave Out*); pino SCK (*Serial Clock*); pino GND (*ground*) e pino +5V.

5 Teste com sensores de mouse óptico

Os testes foram baseados em dois exemplos [8], [9] diferentes, em que os sensores usados foram o PAN-3101DB e o ADNS-2051. Os testes foram realizados com sensores diferentes, os quais foram o ATC-7515DB e o BCE-11341. São sensores de formatos e pinagens diferentes, indicando a existência de vários tipos e a complexidade de funcionamento de cada mouse específico.

Não foi possível encontrar seus *datasheets*, então foram usadas como referência as folhas de dados dos sensores dos exemplos seguidos [10], cada um ao seu semelhante em número de pinos.

Comparando o sensor ATC-7515DB com o PAN-3101DB, além de possuírem o número de referência similares, eles corresponderam a todas as características físicas, propondo assim a nítida possibilidade dos pinos correspondentes terem a mesma função descrita na folha de dados do sensor exemplo.

Analisando também os sensores BCE-11341 com o ADNS-2051, a única semelhança física que possuem é o número de pinos. Dessa forma é provável que a pinagem deles, em termos de função, não correspondam.

5.1 Experiência com ATC 7515DB

Nessa experiência foram utilizados apenas quatro pinos: o pino 3 SDIO (*Serial data (input and output)*); pino 4 SCLK (*Serial port clock (input)*); pino 6 GND (*System ground*); pino 7 PWR (*5V power supply*).

O esquema foi montado da seguinte forma: pino 3 conectado ao pino digital 3 do Arduino; o pino 4 ao pino digital 2; o pino 6 ao GND do Arduino e o pino 7 ao pino 5V. A montagem é vista na figura 1.

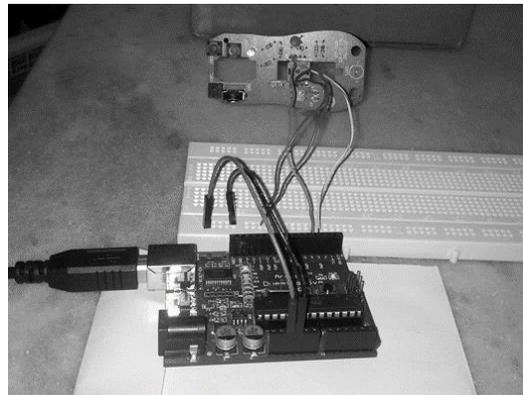


Fig. 1. Montagem do sensor ATC 7515DB.

Porém, ao tentar compilar o programa o mesmo apresentou erro e não foi possível solucionar tal problema. Erro mostrado na figura 2.



Fig. 2. Erro no programa do sensor ATC 7515DB.

5.2 Experiência com o BCE 11341

Com esse sensor utilizamos desta vez cinco pinos: o pino 1 SCLK; pino 12 GND; pino 13 PWR; pino 15 PD (*Power down pin*) e pino 16 SDIO.

Os pinos foram conectados da seguinte forma: pino 1 conectado ao pino digital 5; o pino 12 ao GND do Arduino; o pino 13 ao pino 5V; o pino 15 ao pino digital 7 e o pino 16 ao pino digital 6. A montagem é vista na figura 3.

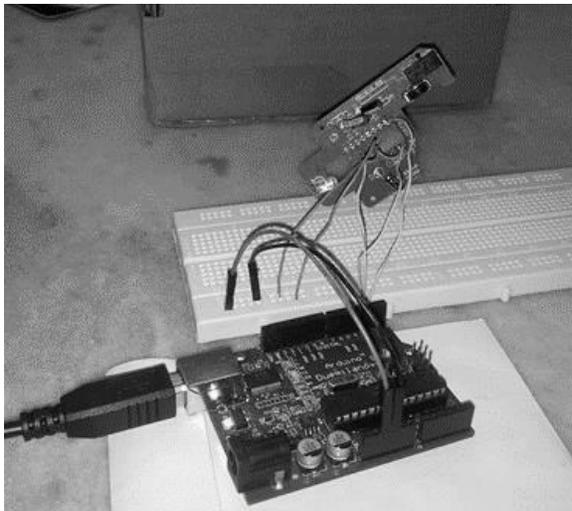


Fig. 3. Montagem do sensor BCE 11341.

O programa foi testado, compilou normalmente e depois foi carregado no Arduino duemilanove. Ao conferir a leitura do sensor pelo *serial monitor*, tela de monitoração da IDE do Arduino, os dados mostrados não corresponderam ao desejado. Provável que a pinagem do sensor utilizado no teste não confira com a do sensor do exemplo. A leitura é mostrada na figura 4.

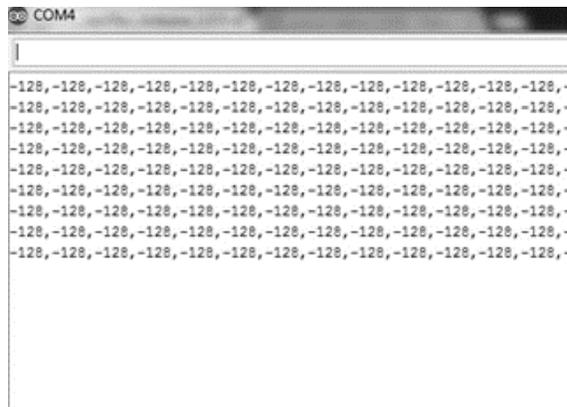


Fig. 4. Leitura do sensor BCE 11341.

6 Teste de Funcionamento do módulo de câmera VC0706 com o Arduino deumilanove

Para o teste da câmera seguiu-se os exemplos demonstrados pelo fabricante, em que a câmera executa duas funções: a *snapshot*, onde a câmera captura uma imagem automaticamente, e a *detecting motion*, em que ela captura uma imagem logo após perceber um movimento.

Os quatro pinos utilizados nesses dois testes são: pino +5V; GND; pino RX (*Serial Data Input*); pino TX (*Serial*

Data Output). Esses pinos são conectados da seguinte maneira: pino +5V no pino 5V do Arduino; pino GND no pino GND; um resistor de 10K Ω é conectado no pino digital 3 e o outro, fazendo um divisor de tensão com o primeiro, vai pro GND, e o RX é conectado no ponto divisor de tensão; o pino TX fica no pino digital 2 do Arduino.

Os pinos do módulo micro-SD são conectados da seguinte forma: pino CS no pino digital 10 do Arduino; pino MOSI no pino digital 11; pino MISO no pino digital 12; pino SCK no pino digital 13; pino 5V nos 5V do Arduino e pino GND no GND do Arduino. A montagem é apresentada na figura 5.

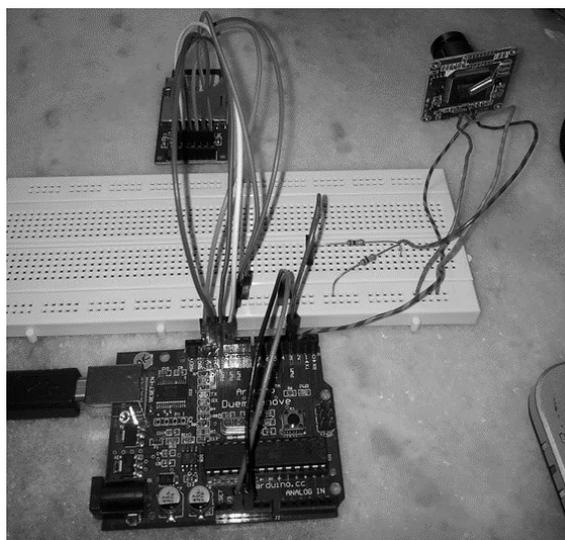


Fig. 5. Montagem da câmera VC0706.

6.1 Teste *snapshot* e teste *detecting motion*

Nesses testes o arduino reconhece a câmera e o adaptador; caso não sejam encontrados por algum motivo, é indicado o problema pelo *serial monitor* da IDE do arduino.

Na execução do teste de *snapshot* o programa nos informa a resolução da imagem, que é no máximo de 640x480 pixels, informa que a foto foi capturada, o tamanho da imagem e o tempo que levou para salvar a imagem. Isso pode ser visto na figura 6. E na figura 7 a imagem capturada na realização do teste.

```

COM4
VC0706 Camera snapshot test
Camera Found:
-----
VC0703 1.00
Ctrl infr exist
User-defined sensor
525
-----
Image size: 640x480
Snap in 3 secs...
Picture taken!
Storing 46896 byte image.....done!
13582 ms elapsed
    
```

Fig. 6. Informações do teste *snapshot*.

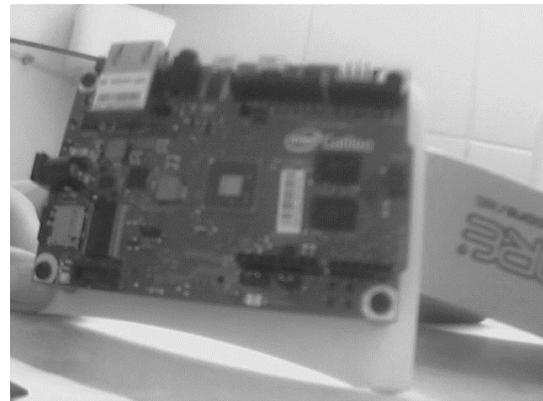


Fig. 9. Imagem capturada pela câmera.

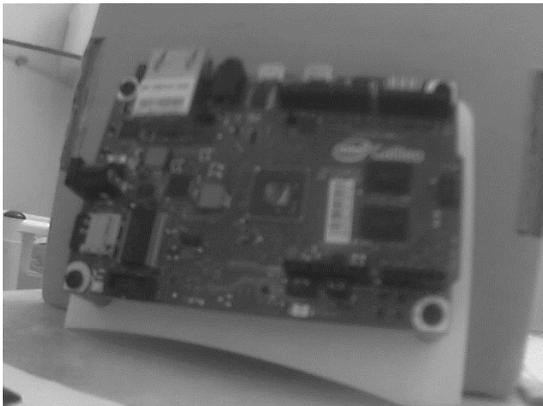


Fig. 7. Imagem capturada pela câmera.

No *detecting motion* o programa nos informa a resolução da imagem, informa que a função *motion detect* está habilitada, e quando a câmera percebe um movimento ela acusa e captura uma imagem. É informado tamanho da imagem e o tempo que levou para salvá-la no cartão de memória. Pode ser visto na figura 8 tais informações e na figura 9 a imagem capturada assim que a câmera detectou um movimento à sua frente.

```

COM4
VC0706 Camera test
Camera Found:
-----
VC0703 1.00
Ctrl infr exist
User-defined sensor
525
-----
Image size: 320x240
Motion detection is ON
Motion!
Picture taken!
12348 byte image
Writing image to IMAGE01.JPG...Done!
Motion!
Picture taken!
12088 byte image
Writing image to IMAGE02.JPG...Done!
    
```

Fig. 8. Informações do teste *detecting motion*.

7 Teste de funcionamento do módulo de câmera VC0706 com o Arduino galileo

O galileo foi incorporado ao trabalho por ser um arduino e devido ao seu alto poder de processamento em relação aos outros modelos; o que seria ideal já que processamento de imagem demanda memória e um bom processador.

Devido à possível comunicação do módulo de câmera com o Arduino duemilanove, tentou-se também com o galileo. Aplicando os mesmos testes de funcionamento da câmera realizados com o duemilanove, as conexões dos componentes foram realizadas da mesma forma já que o galileo possui a pingem igual às do uno e do duemilanove.

Após instalar a IDE do galileo, uma IDE própria, pois ele não usa a IDE comum do Arduino, a mesma não inicializa normalmente, devido a um problema em que a IDE só funciona se o idioma do formato da data e hora do computador for o inglês. A montagem pode ser vista na figura 10.

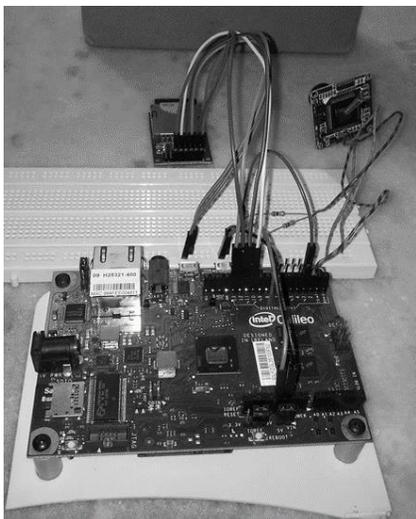


Fig. 10. Montagem da câmera com o galileo.

O programa não compilou apresentando erros diversos, apontando declarações não realizadas no programa. Chegou-se a conclusão de que a biblioteca da IDE do galileo não possuía a documentação necessária para que o programa compilasse. O que não aconteceu com a IDE comum do Arduino. Tais problemas de compilação já são de conhecimento dos desenvolvedores da Intel que estimam as correções para a próxima versão da IDE. O erro pode ser visto na figura 11.

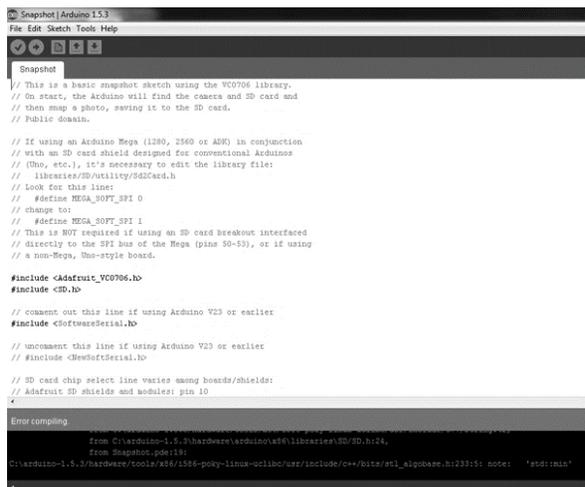


Fig. 11. Erro na IDE do galileo.

8 Conclusões

Este trabalho apresentou a tentativa de aplicar o Arduino em visão estereoscópica; tentativa sem sucesso devido à falta de referências de trabalhos sobre Arduino com estereoscopia e também ao limitado conhecimento na sua linguagem de programação, pois os programas referentes à comunicação da câmera com o Arduino são bem complexos.

Os testes com os sensores de mouse óptico, mesmo mostrando a possibilidade de sua comunicação com o Arduino, não corresponderam iguais aos exemplos de referência. De toda forma, se entende como inviável a utilização deles para processo estereoscópico, já que seus alcances ópticos são bem limitados.

A câmera VC0706 se apresentou como acessório ideal para trabalhar com o Arduino, visto que os testes de funcionamento foram bem sucedidos. Porém não foi possível verificar se eles têm condições de trabalharem com estereoscopia, pelos motivos citados anteriormente.

Acredita-se que com maior conhecimento na linguagem de programação, juntamente com o *datasheet* da câmera, seja possível implementar um programa em que o Arduino comande a câmera e processe as imagens capturadas, podendo calcular a distância do objeto para as câmeras.

Referências

- [1]] G.R.P. Esteves, Estereoscopia no cálculo de distância e controle de plataforma robótica, UPE, Recife, 2011.
- [2] Santos, N.P., Arduino - Introdução e Recursos Avançados, Escola Naval, 2009.
- [3] Di Renna, R.B., Brasil, R.D.R., Cunha, T.E.B., Beppu, M.B., Fonseca, E.G.P., Introdução ao kit de desenvolvimento Arduino, UFF, Niterói - RJ, 2013.
- [4] Site da Internet: www.arduino.cc., visitado pela última vez em 12/09/2014.
- [5] Figueira, J.S., Einstein, Mouse Ótico e Maisena – Uma Análise do Movimento Browniano, UTFPR, Pato Branco – PR.
- [6] Borges, J.C.N., MouseCAM, Curitiba, 2008.
- [7] Site da internet: www.adafruit.com, visitado pela última vez em 12/092014.
- [8] Site da Internet: www.martijnthe.nl/2009/07/interfaces-an-optical-mouse-sensor-to-your-arduino/., visitado pela última vez em 11/09/2014.
- [9] Site da internet: www.frenki.net/2013/12/convert-optical-mouse-into-arduino-web-camera/., visitado pela última vez em 11/09/2014.
- [10] Site da internet: www.alldatasheet.com/., acessado pela última vez 22/06/2014.