

Apesar do Scratch possuir alguns métodos de realizar a conexão com o Arduino, a aplicação da PixyCam utilizando a placa junto com a ferramenta de desenvolvimento se torna aparentemente inviável. A interação realizada entre a placa e o Scratch ocorre quase que totalmente no controle dos pinos lógicos e analógicos, sendo que a proposta da conexão entre o Scratch e o Arduino seria executar o controle desses pinos sem usar linhas de código.

A programação necessária para a utilização das funcionalidades de reconhecimento da câmera é bem específica, apesar de ser extremamente simples. Os códigos são feitos dentro da IDE do Arduino e carregados para placa. A seguir temos alguns exemplos de comandos para a utilização da PixyCam no Arduino:

- `pixy.ccc.grabBlocks()` Retorna os objetos reconhecidos;
- `pixy.ccc.blocks[i].m_signature` Retorna a assinatura (o tipo do objeto definido nas configurações de reconhecimento da câmera) do objeto rastreado;
- `pixy.ccc.blocks[i].m_x` Retorna a posição x do objeto rastreado;
- `pixy.ccc.blocks[i].m_y` Retorna a posição y do objeto rastreado;
- `pixy.ccc.blocks[i].m_width` Retorna a largura do objeto rastreado;
- `pixy.ccc.blocks[i].m_height` Retorna a altura do objeto rastreado;
- `pixy.ccc.blocks[i].m_angle` Retorna o ângulo do objeto rastreado;
- `pixy.ccc.blocks[i].m_index` Retorna o index do objeto rastreado;
- `pixy.ccc.blocks[i].m_age` Retorna o número de frames que o objeto está sendo rastreado;
- `pixy.ccc.blocks[i].print()` Exibe as informações do objeto.

A utilizações da biblioteca Pixy para interação com a PixyCam2 é bem simples e intuitiva. Porém, a utilização dessa biblioteca em qualquer um dos métodos de conexão entre o Arduino e o Scratch é aparentemente impossível, pois, não há como inserir linhas de código, muito menos importar a biblioteca para podermos utilizar os comandos. Portanto, apesar da conexão funcionar e ser possível utilizar dela para diversos fins, ela interage mais diretamente com as portas e não permite uma abrangência maior no uso da placa, o que inviabiliza o uso de tais métodos.

O principal problema em utilizar dos métodos antes mencionados (ScratchX, S4A e HackEduca Conecta) para a conexão do Arduino com o Scratch é que não podemos passar variáveis do Arduino para os blocos do Scratch. Para entendermos melhor é preciso explicar o que a placa realmente faz ao interagir com a câmera e todo o funcionamento em si.

A PixyCam, ao ser conectada ao computador (para o funcionamento da câmera, esta precisa estar conectada tanto na placa quanto no computador) interage com um programa chamado PixyMon. Este programa é responsável pelas configurações da câmera, e exibe na tela as imagens que estão sendo registradas pela mesma. Através desse programa podemos fazer a configuração da câmera, definindo a luminosidade, quantidade de frames por segundo, quantidade de pixels a serem analisados, qualidade da imagem, dentre diversos outros ajustes para definir um melhor reconhecimento dos objetos. Além dessas configurações, o PixyMon também faz a definição dos objetos a serem reconhecidos, definindo-os com assinaturas utilizadas para identificar os objetos.

A interação entre a câmera e a placa é bem simples. Construímos o código desejado dentro da IDE do Arduino carregando os plugins necessários para dentro do software (todos os plugins, firmware, e softwares necessários são disponibilizados no site da PixyCam) e carregamos os códigos para dentro da placa. Como dito anteriormente, a programação da placa para interagir com a câmera é bem simples e são necessários poucos comandos para conseguir as informações dos objetos reconhecidos. A ideia da interação entre a placa e a câmera é coletar as informações dos objetos reconhecidos pela PixyCam. Tais informações são controladas pelo PixyMon, que é a ponte entre a câmera e a placa.

Resumidamente então temos que, conectamos a câmera no Arduino e no computador. A câmera interage com um programa que controla o reconhecimento dos objetos, as configurações da câmera, e as assinaturas dos objetos reconhecidos. Os códigos na placa acessam as informações de posicionamento, assinatura, e demais dados dos objetos reconhecidos, e trabalha em cima dessas informações dependendo dos métodos implementados no código. A seguir temos um exemplo de código:

```

#include <Pixy2.h>

// Criamos o objeto da PixyCam
Pixy2 pixy;

void setup(){
    // Definimos a saída no terminal
    Serial.begin(115200);

    // Iniciamos a câmera
    pixy.init();
}

void loop(){
    int i;

    // Pega os objetos detectados
    pixy.ccc.getBlocks();

    // Se existirem objetos reconhecidos eles são exibidos no
    terminal
    if (pixy.ccc.numBlocks){
        Serial.print("Detectado: ");
        Serial.println(pixy.ccc.numBlocks);
        for (i = 0; i < pixy.ccc.numBlocks; i++){
            Serial.print("\n Bloco ");
            Serial.print(i);
            Serial.print(": ");

            // Exibe todas as informações do objeto detectado
            pixy.ccc.blocks[i].print();
        }
    }
}

```

Código 1 Exibindo os objetos reconhecidos

A conexão entre a placa e a câmera se dá de uma maneira bem específica através dos códigos, e esta interação não é possível pelos métodos convencionais de interação disponíveis anteriormente citados, que tratam da interação em um âmbito mais geral.