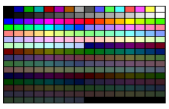
E-mail enviado
Fernando Simões Garcia
0 Comentários

Explorando o módulo OV7670-FIFO: Interface SCCB



ÍNDICE DE CONTEÚDO

- Módulo da câmera VGA
- Interface de Comunicação SCCB
- Registradores de Controle
  - Ciclo de escrita dividido em três fases
  - Ciclo de escrita dividido em duas fases
- Próximo artigo
- Bibliotecas

Este post faz parte da série Explorando o módulo OV7670-FIFO. Leia também os outros posts da série:

- Explorando o módulo OV7670-FIFO: Interface SCCB
- Explorando o módulo OV7670-FIFO: Video Frame Buffer

Olá, caro leitor. Neste artigo, apresentarei os procedimentos necessários para acessar o módulo da câmera VGA OV7670. O artigo será dividido em duas partes, sendo a segunda referente ao processo para capturar imagens.

Módulo da câmera VGA

O módulo da câmera VGA utiliza o sensor de CMOS OV7670. As principais características desse sensor são indicadas a seguir [1]:

- Resolução máxima: 640 x 480;
- Taxa de transferência: Máxima de 30 fps em modo VGA;
- Tensão de operação: 2.5 V à 3.0 V;
- Alta sensibilidade em ambientes com pouca luz;
- Interface SCCB (Serial Camera Control Bus), compatível com interface I2C;
- Suporta VGA, CIF, ~ Formatos de saída: YUV/YCbCr 4:2:2, RGB 565/555, GRB 4:2:2, RAW RBB Data;
- Controle automático de funções de imagem;
- Controle automático de qualidade de imagem, como sobrancia, matiz, gama, nível e anti-blooming;
- Auto detecção de flicker (50/60 Hz);
- Tamanho da lente: 1/6 ";
- Ângulo de visão: 24 graus.

As características descritas são referentes ao sensor de imagem. Outras informações podem ser obtidas tanto no Application Note quanto no Datasheet do OV7670.

Em relação ao módulo, duas versões podem ser encontradas. A primeira utiliza somente o sensor OV7670 e apresenta os pinos para conexão entre o sensor e outro dispositivo. Essa versão é mostrada na Figura 1.

Figura 1. Câmera VGA – OV7670 [2].

A pinagem do módulo é mostrada na Figura 2.

Figura 2. Pinagem do módulo da câmera VGA [3].

Já na segunda versão, um buffer FIFO – First In First Out) é utilizado para armazenar os dados do sensor CMOS. O dispositivo utilizado é o CI AL422 projeto para tratar sinais de vídeo. Essa versão é mostrada nas Figura 3 e 4.

Figura 3. Módulo da câmera VGA + FIFO – Frente.

Figura 4. Módulo da câmera VGA + FIFO – Verso.

A pinagem do módulo é mostrada na Figura 5.

Figura 5. Pinagem do módulo da câmera VGA + FIFO.

A diferença da pinagem entre os módulos é que o segundo apresenta sinais de controle do módulo FIFO.

Interface de Comunicação SCCB

Independente do módulo, a interface de comunicação com o sensor é a mesma. Tal interface é chamada de SCCB e é compatível com o protocolo I2C. Assim, dois pinos são utilizados na comunicação, sendo um para a sincronização e outro para transferência de dados. Essa configuração é ilustrada na Figura 6.

Figura 6. Diagrama SCCB [3].

Para estabelecer a interface SCCB foram definidas cinco funções:

- SCCB_Init: Determina condição inicial dos sinais de controle da interface SCCB;
- SCCB_Start: Inicia comunicação;
- SCCB_Stop: Finaliza comunicação;
- SCCB_ReadByte: Faz a leitura de um byte;
- SCCB_WriteByte: Escreve um byte.

Tais funções foram declaradas no arquivo de cabeçalho SCCB.h. Cabe ressaltar que o projeto pode ser encontrado neste link.

```
#ifndef SCCB_H
#define SCCB_H

#include "Arduino.h"

#define SIOC_PIN 2
#define SIOC_PIN_MASK (1<<2)
#define SIOC_DDR_REGISTER(x) DDRD
#define SIOC_PORT_REGISTER(x) PORTD
#define SIOC_OUTPUT(x)
#define SIOC_HIGH(x)
#define SIOC_LOW(x)
#define SIOD_PIN 3
#define SIOD_PIN_MASK (1<<3)

#endif
```

O sinal SIO_C é controlada pelo dispositivo mestre, indicando a transferência de 1 bit quando o sinal passa do nível lógico 0 para 1. Quando o transmissor está ocioso, o sinal SIO_C deve permanecer em nível lógico 1. Assim, quando uma transferência é iniciada, o sinal SIO_C é colocado em nível lógico 0. Convenientemente, a condição dos sinais é definida na função SCCB_INIT.

```
void SCCB_Init(void)
{
  //Serial Clock (SIOC) and Data (SIOD) are output pins and determines the initial condition
  SIOC_OUTPUT(1);
  SIOC_HIGH(1);
  SIOD_OUTPUT(1);
  SIOD_HIGH(1);

  //Delay 1 cycle to finish update
  SIOC_OUTPUT(1);
  SIOC_LOW(1);
  SIOD_OUTPUT(1);
  SIOD_LOW(1);
}
```

O início da transmissão é dado quando o sinal SIO_D passa de nível lógico 1 para 0, e na ampliação o sinal do clock também vai a zero. Já o final da transmissão ocorre quando o sinal SIO_D é posto em nível lógico 1 enquanto o sinal SIO_C está em nível lógico 1. Essas duas situações são ilustradas na Figura 7.

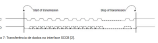


Figura 7. Transferência de dados na interface SCCB [3].

```
void SCCB_StartCond(
  //Serial Data (SIOD) on the SDA and SCL Clock
  SIOD_OUTPUT(1);
  SIOD_LOW(1);

  //Delay 1cycle to set the SDA line
  SIOD_OUTPUT(1);

  //Delay 1cycle to set the SCL line
  SIOC_OUTPUT(1);
  SIOC_LOW(1);
}

void SCCB_StopCond()
{
  //Serial Data (SIOD) OFF the SDA and SCL Clock
  SIOD_OUTPUT(1);
  SIOD_HIGH(1);

  //Delay 1cycle to the SCL line
  SIOC_OUTPUT(1);
  SIOC_HIGH(1);

  //Delay 1cycle to the SDA line
  SIOD_OUTPUT(1);
  SIOD_LOW(1);
}
```

Essas funções são usadas pelo módulo OV7670 para as operações de escrita e leitura dos registradores de controle. Para tal, foram definidas as funções de transferência de dados entre os dispositivos.

```
void SCCB_WriteByte(uint8_t data)
{
  uint8_t i;
  //Delay 1 cycle to stay stable
  //Clock is not yet low (begin the low time to the corresponding value)
  for (i=0; i<8; i++)
  {
    //Send a bit on the port
    if(data & 0x80) {
      SIOD_OUTPUT(1);
      SIOD_HIGH(1);
    }
    else {
      SIOD_OUTPUT(1);
      SIOD_LOW(1);
    }

    //Delay 1 cycle to stay active
    SIOC_OUTPUT(1);
    SIOC_HIGH(1);
    SIOC_OUTPUT(1);
    SIOC_LOW(1);

    //Move to next bit
    data = data << 1;
  }

  SIOD_PIN_MODE(1);

  //Generate a clock pulse
  SIOC_OUTPUT(1);
}
```

Registradores de Controle

Com a interface SCCB implementada é possível configurar os registradores de controle do chip para o modo de operação desejado. Essas configurações são realizadas a partir de uma sequência comta de operações, chamadas ciclos de transmissão, que são classificadas da seguinte maneira:

- Ciclo de escrita dividido em três fases;
- Ciclo de escrita dividido em duas fases;
- Ciclo de leitura dividido em duas fases.

De modo geral, cada fase de transmissão envolve a transferência de 9 bits de dados e um nano bit, sendo função **Don't Care**. Além disso, na primeira fase da transmissão ocorre a escrita do endereço do dispositivo no barramento SCCB.

Ciclo de escrita dividido em três fases

Esse ciclo é utilizado para escrever em um registrador do dispositivo. Assim, a segunda informação transferida corresponde ao endereço do registrador seguido do valor que será armazenado.

Figura 8. Ciclo de escrita dividido em três fases.

```
void SCCB_WriteReg(uint8_t id, uint8_t, uint8_t data)
{
  //Send ID address
  SCCB_Start();

  SCCB_WriteByte(id);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Send sub-address
  SCCB_WriteByte(sub);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Send data
  SCCB_WriteByte(data);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Stop
  SCCB_Stop();
}
```

Ciclo de escrita e leitura dividido em duas fases

Esse procedimento é utilizado para indicar o endereço de um registrador que será acessado pela função de leitura. Cabe ressaltar que essa etapa ocorre no ciclo de escrita dividido em três fases.

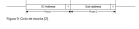


Figura 9. Ciclo de escrita [3].

Para ler o valor de registrador é necessário determinar novamente o ID do dispositivo e, em seguida, capturar o valor.

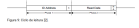


Figura 10. Ciclo de leitura [3].

```
uint8_t SCCB_ReadReg(uint8_t id,uint8_t, uint8_t);
{
  //Send ID address
  SCCB_Start();

  //Send ID address
  SCCB_WriteByte(id);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Send sub-address
  SCCB_WriteByte(sub);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Stop
  SCCB_Stop();

  //Start
  SCCB_Start();

  //Send ID address
  SCCB_WriteByte(id);
  SCCB_WriteNinthBit(0);
  SCCB_WriteNinthBitDontCare();

  //Read data
  uint8_t data = SCCB_ReadByte();
  SCCB_WriteNinthBit(1);
  SCCB_WriteNinthBitDontCare();

  //Stop
  SCCB_Stop();

  return data;
}
```

Próximo artigo

No próximo artigo será apresentado o procedimento para capturar um frame do módulo OV7670 utilizando o buffer AL422. Além do controle o buffer, a interface SCCB será utilizada para realizar a configuração dos registradores de controle do módulo OV7670.