



PS/2 Keyboard for MSX And Legacy computers

**Based on ARM Cortex M3
Blue Pill STM32F103C8T6**

Introdução

O principal objetivo foi aprender a usar o Ecosistema de Desenvolvimento aberto (free) ARM Core M3

Para tal, escolhi uma aplicação de missão crítica, implementando facilidades de configuração normalmente não observadas nas várias soluções inteligentes já implementadas, como, por exemplo, permitir a atualização da tabela de conversão sem a necessidade e de recompilar o código.

Solução simples e de baixo custo

Índice

Considerações e características de
contorno dos ambientes

Escolha do MCU

Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

Assunto do momento

Considerações e características de
contorno dos ambientes

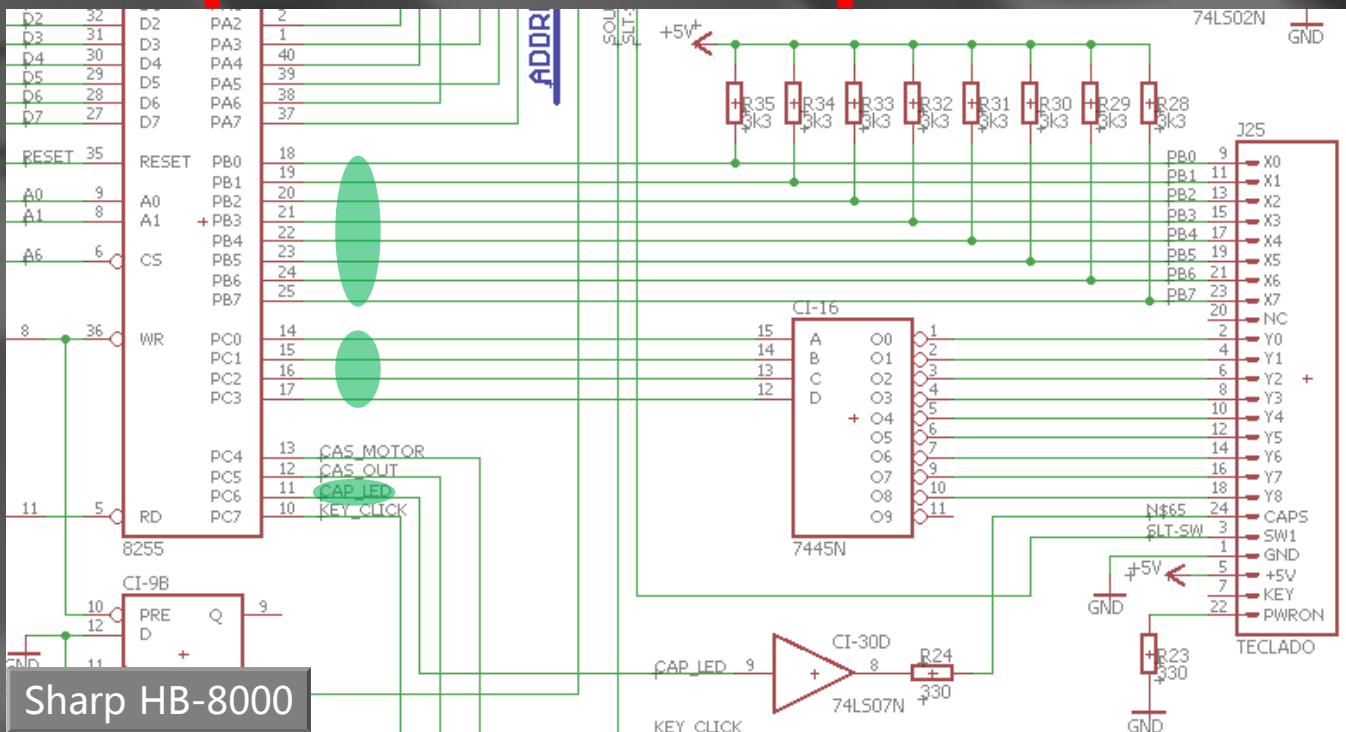
Escolha do MCU

Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

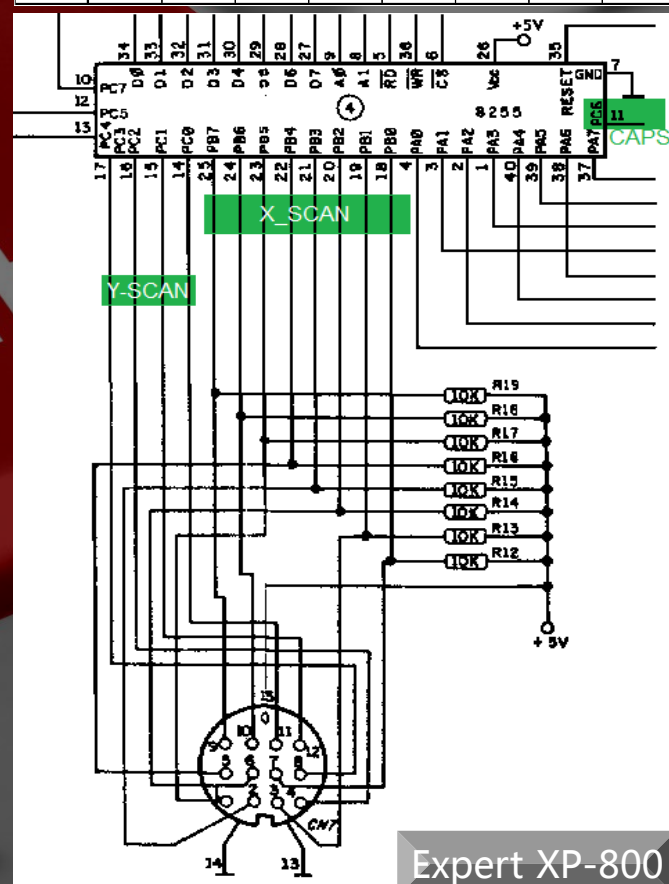


O teclado MSX é uma matriz de 8 linhas por até 15 colunas, o que nos leva a um máximo teórico de 120 teclas – O HB8000 tem 9 colunas. MSX japoneses highend usam teclados com 11 colunas;

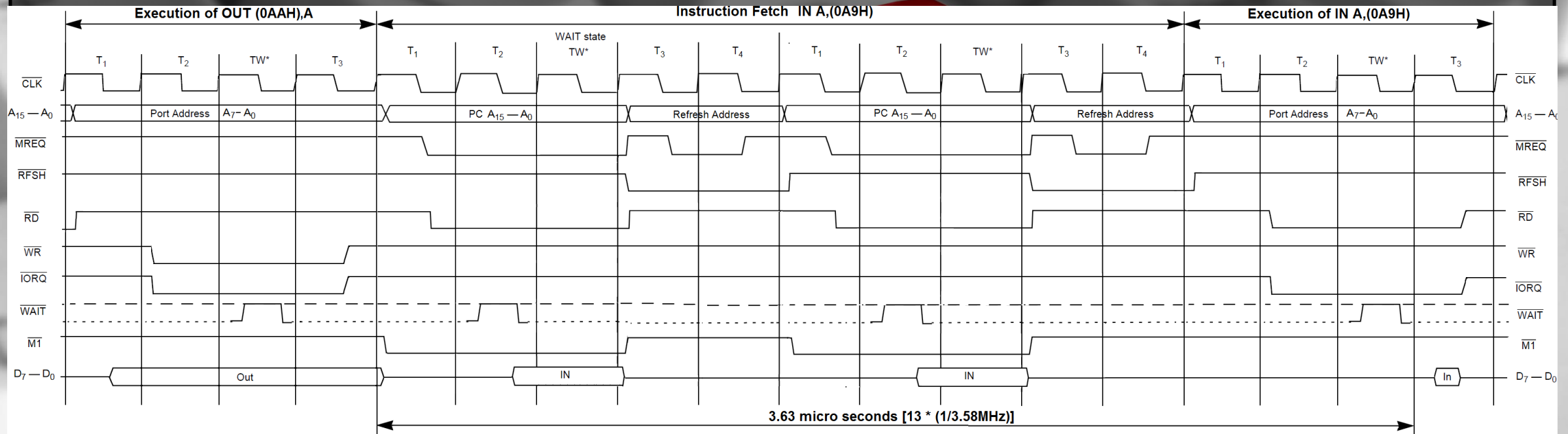
Primeiramente, o scan é feito através de uma escrita em Y3:0 (PC3:0, endereço 0AAH), que vai para um decoder até 4x16 com Open-collector outputs, e lidas as linhas X7:0 (através da PPI 8255 PB7:0, port 0A9H);

Os estados do Caps Lock e Katakana (ativos em "0") são lidos diretamente do hardware do MSX.

	X = 0	X = 1	X = 2	X = 3	X = 4	X = 5	X = 6	X = 7
Y = 0	0	1	2	3	4	5	" 6	7
Y = 1	8	9	- =	+ =	^ \	' ..	' ..	Ç
Y = 2	^ ~	[]	; ,	: .	? /	> <	A	B
Y = 3	C	D	E	F	G	H	I	J
Y = 4	K	L	M	N	O	P	Q	R
Y = 5	S	T	U	V	W	X	Y	Z
Y = 6	SHIFT	CTRL	GRAPH	CAPS	CODE	F1	F2	F3
Y = 7	F4	F5	ESC	TAB	STOP	BackSpace	SELECT	RETURN
Y = 8	Space	CLS HOME	INS	DEL	LEFT	UP	DOWN	RIGHT



Temporização



Mapeamento detalhado da temporização de leitura de teclado MSX (OUT seguido de IN)

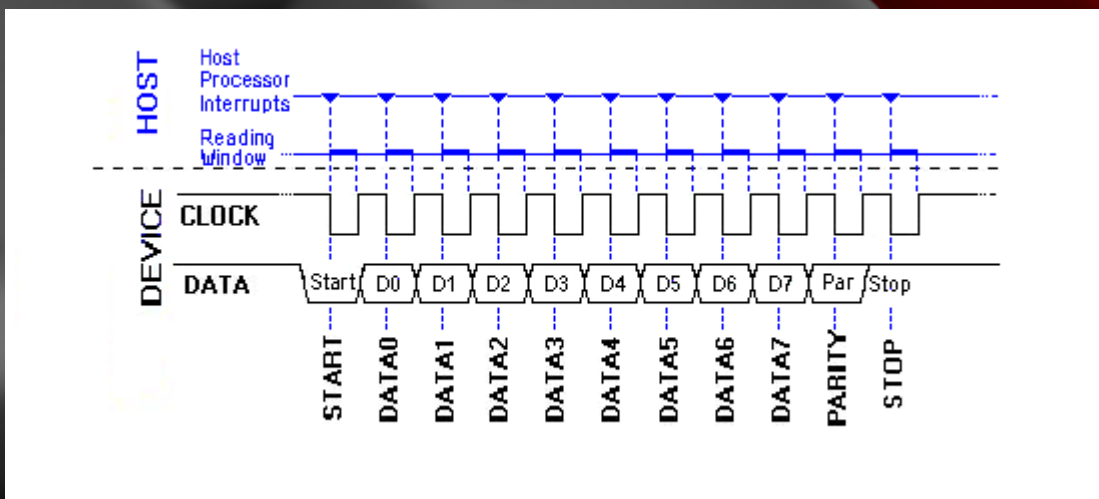
- O tempo após a atualização do Y (OUT (0AAH), A) disponível para responder com a linha (X7:0) é de 3,6μs;
- A leitura de teclado (IN A, 0A9H) não é feita somente após o OUT (exemplo: jogos), portanto, o adaptador deve estar preparado para atualizar X, para o respectivo Y que estiver registrado no momento, de acordo com as atualizações do teclado PS/2.

O Teclado PS/2 - Temporização

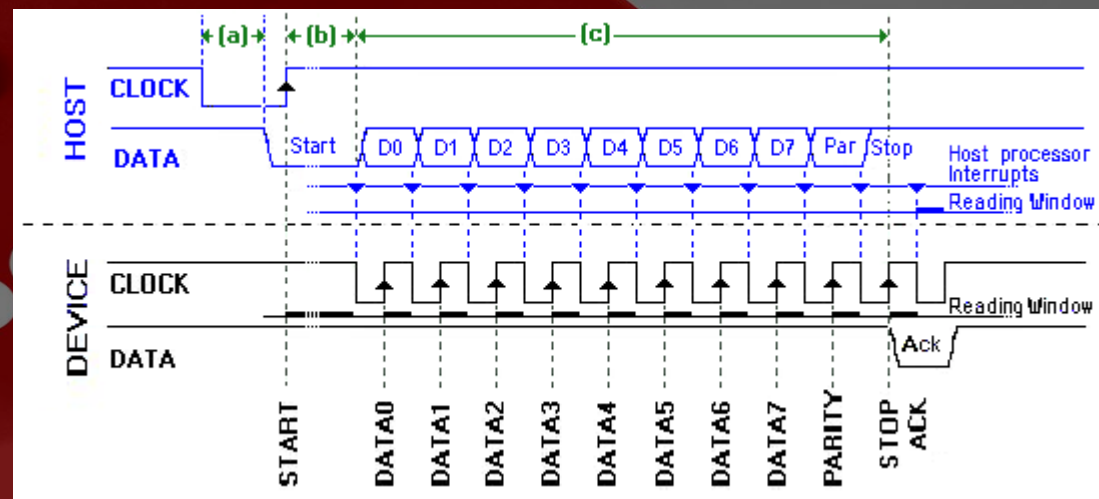
- O teclado PS/2 se comunica através de uma comunicação serial síncrona *half duplex* de dois fios: Dados e Clock;
- A velocidade do clock varia entre 10K e 16,7KHz;

Seguem diagramas de tempo detalhado para:

1) Comunicação Teclado para o host (PC ou adaptador):



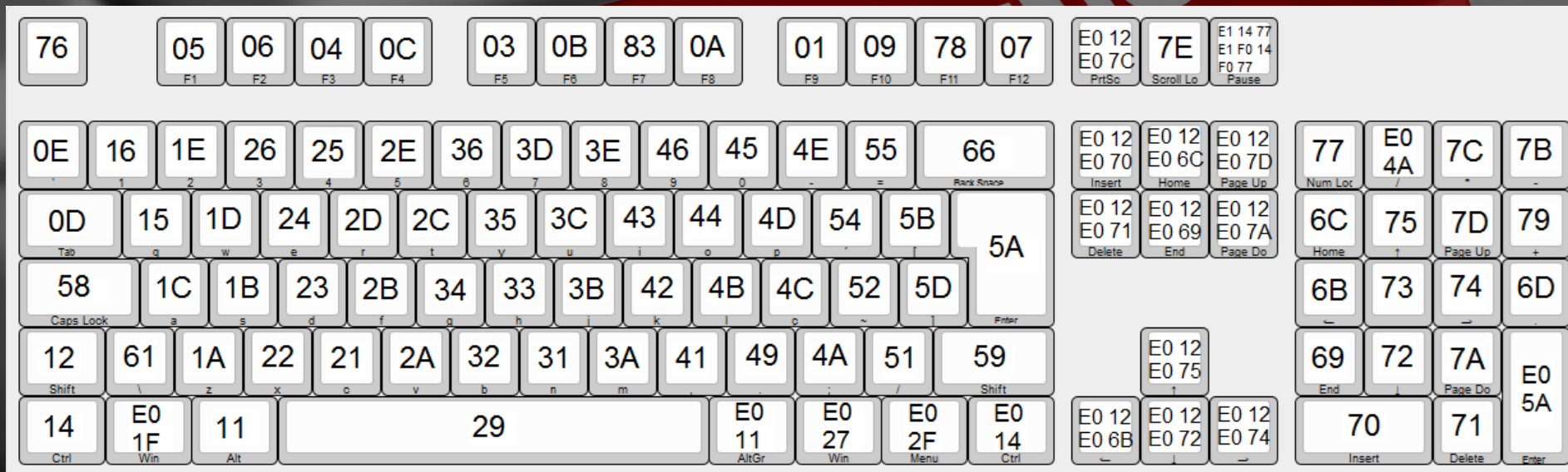
2) Envio de comandos do Host para o teclado



- (a) Host força o clock nível baixo (0) por pelo menos 100µs;
- (b) É o tempo que o teclado requer para enviar clocks para receber o frame, e pode levar até 15ms;
- (c) É o tempo para concluir a recepção do frame, após o teclado iniciar o envio do clock, e deve ser até 2ms;

O Teclado PS/2 - Layout

- Há várias teclas exclusivas de MSX, como Graph, Code, Select e Stop. Tal raciocínio vale também para o teclado PS/2;
- Os scan codes não têm relação com as funções indicadas nos keytops;
- Os scan codes têm tamanhos que variam de 1 a 8 bytes;
- O teclado PS/2 envia códigos de scan tipo 2 para o host e auto-repeat da última tecla pressionada;
- Os códigos de scan são dependentes do layout físico – Por exemplo: Os códigos 51 (tecla / ao lado do Shift direito) e 6D (tecla . do teclado numérico reduzido) são exclusivos do teclado ABNT2 (Id=275);
- Segue exemplo do teclado base para este desenvolvimento (Teclado ABNT2) com os scan codes de make (pressionar):



Assunto do momento

Considerações e características de
contorno dos ambientes

Escolha do MCU

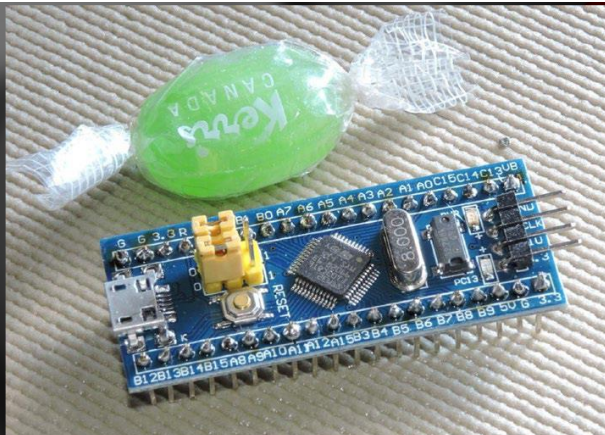
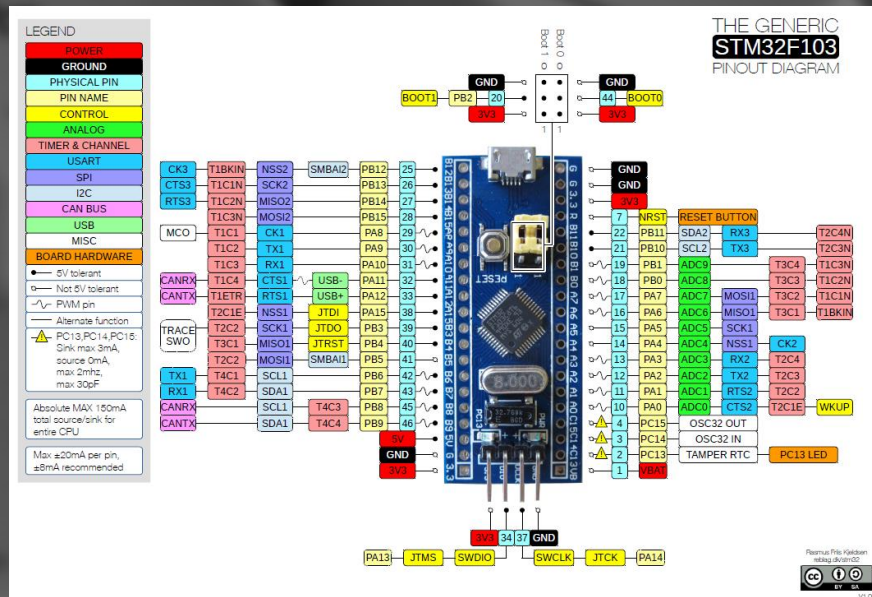
Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

Escolha do MCU (MicroController Unit)



Características atendidas

1. Baixo consumo de energia;
2. 16 Portas capazes de suportar 5V;
3. Portas de saída devem ser programáveis como OD (Open Drain);
4. Portas OD devem poder ter seu estado real lido a qualquer tempo;
5. Pelo menos 7 pinos para interrupção externa;
6. Os pinos de interrupção externa conectados ao Y_SCAN não devem compartilhar recursos de interrupção;
7. Deve ter capacidade de reprogramar a flash através do firmware;
8. Documentação abundante;
9. Ecossistema de desenvolvimento gratuito;
10. Baixo custo;
11. Ser de fácil aquisição.

Assunto do momento

Considerações e características de
contorno dos ambientes

Escolha do MCU

Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

Setup do ambiente de desenvolvimento

Linux Ubuntu 20.04

ARM GCC Tools

LibOpenCM3

OpenOCD

Visual Studio Code

Os passos iniciais foram dados com a orientação do excelente livro:

Warren Gay, Beginning STM32 Developing with FreeRTOS, libopencm3 and GCC,
que me foi apresentado por Ismael Lopes da Silva, site
<https://www.embarcados.com.br/serie/programacao-com-a-placa-blue-pill/>, a
quem sou muito grato!

Assunto do momento

Considerações e características de
contorno dos ambientes

Escolha do MCU

Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

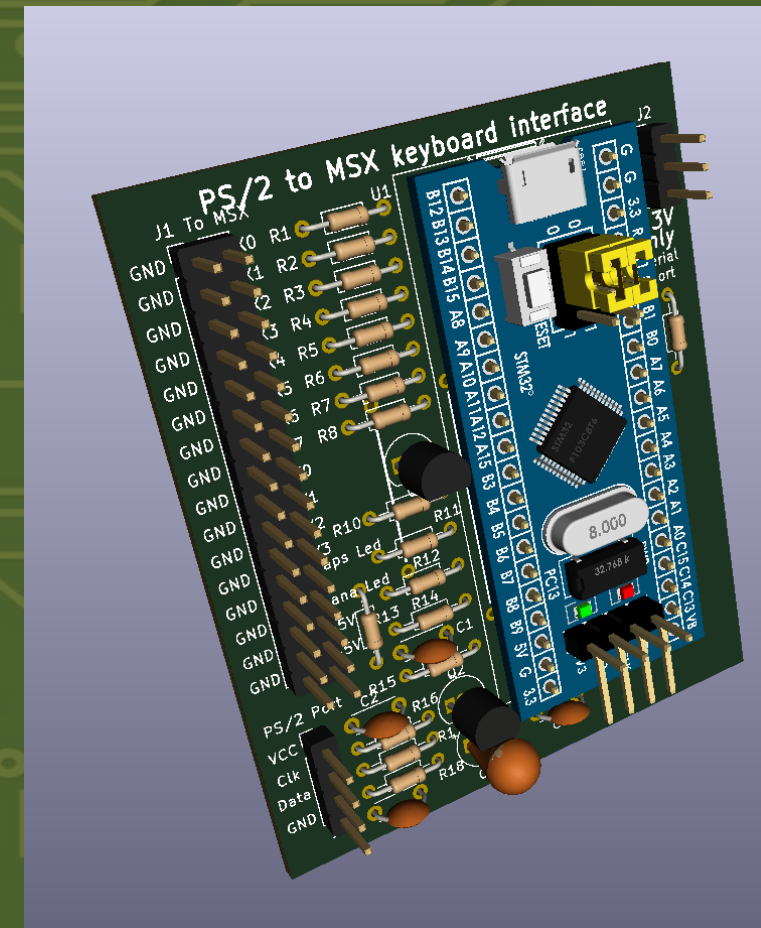
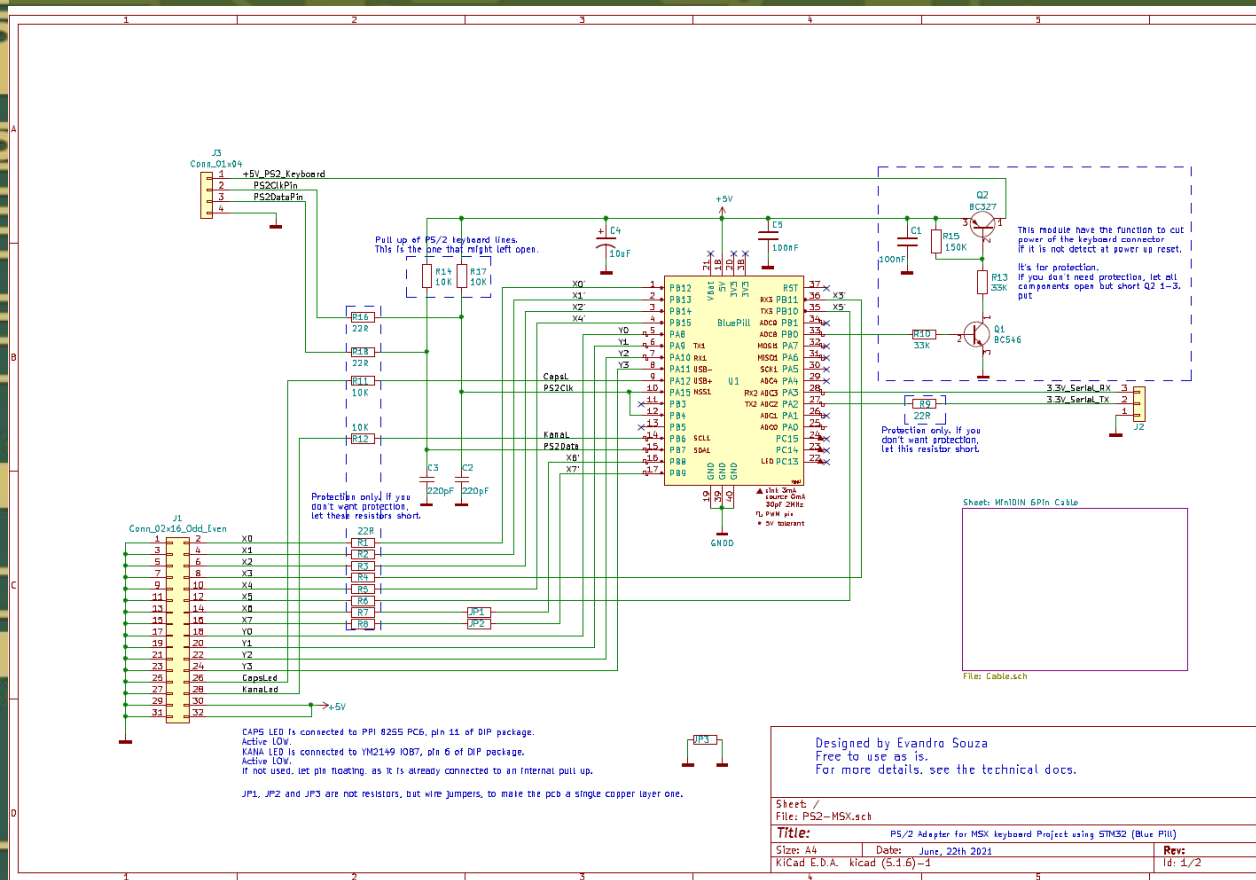
Descrição técnica do firmware

A Eletrônica

O uso do módulo traz as vantagens:

- Alimentação 3,3V (nativa) e 5V do MSX hospedeiro;
- Já vem com os circuitos de apoio (reset, cristais, jumpers de pgm e status led);
- DIP com passo 2,54mm, ao invés de SMD

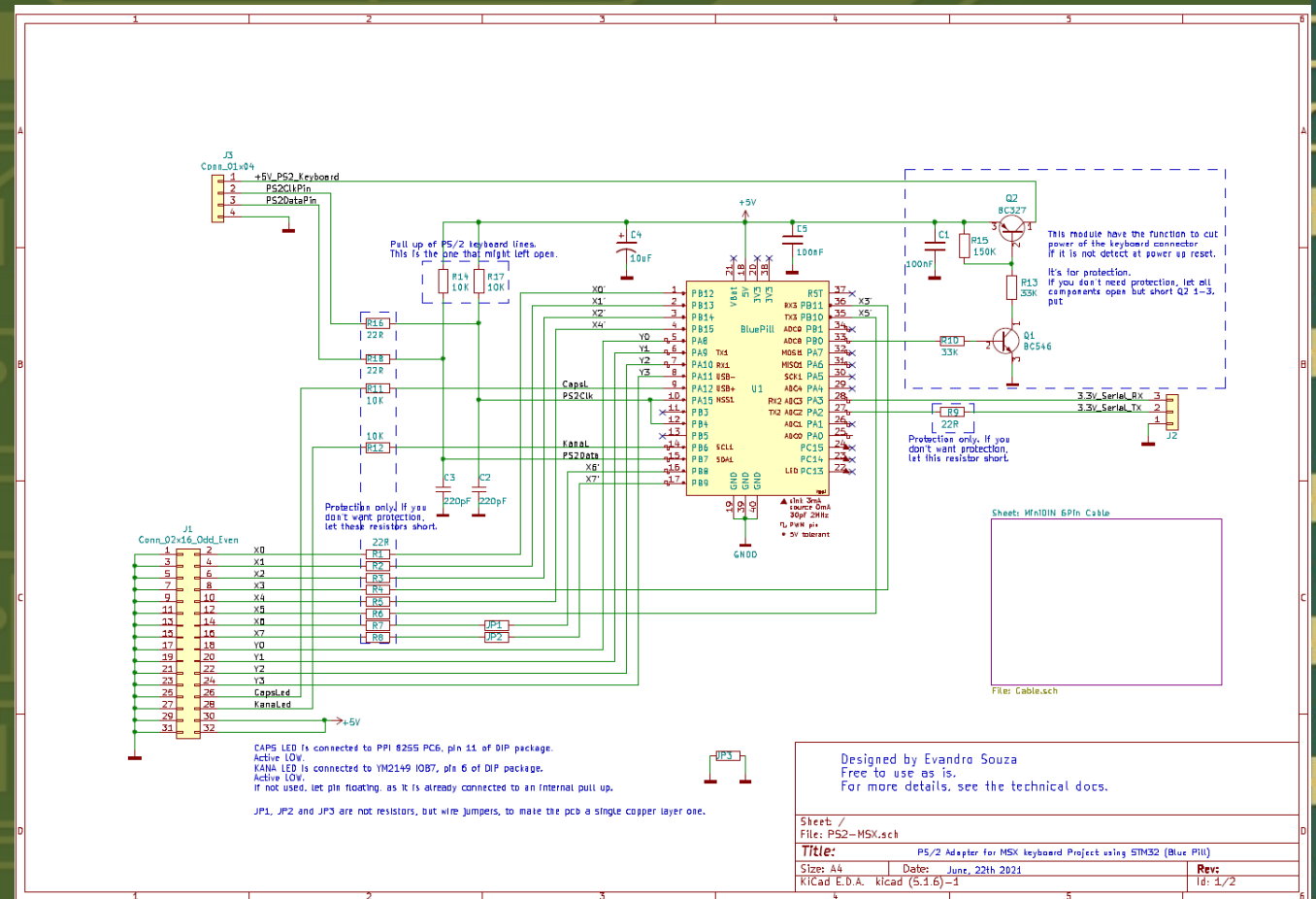
- O EDA (Electronic Design Assitant) usado para documentar o desenvolvimento foi o KiCad:
- Tanto Eeschema quanto o Pcbnew



A eletrônica

100% da eletrônica externa ao módulo tem por função a proteção do teclado/módulo/MSX hospedeiro;

- Desenergizar o teclado quando o mesmo não for detectado no boot (start-up);
- C2 e C3 absorvem os picos de chaveamento, diminuindo desta forma, riscos de falsa detecção das informações do teclado PS/2.
- O consumo medido é de 23mA @ 5V;
- Dimensões da PCB de face simples: 65 x 55mm;
- Embora o STM32 seja capaz de atuar como um dispositivo USB, não sobraram recursos para implementação futura de comunicação via USB. Até a comunicação serial foi atendida através de portas de 3,3V (não tolerantes a 5V).



Assunto do momento

Considerações e características de
contorno dos ambientes

Escolha do MCU

Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

Facilidades do software

O mapeamento (Database) de ambos os teclados pode ser trocado a qualquer tempo: Basta colocar um terminal capaz de enviar arquivo texto com a configuração de serial 38400, 8 n, 1, ligar a este módulo sem o teclado PS/2 e seguir as instruções: Uma operação de 15 segundos.

O Database é alterado via um arquivo Intel Hex, montado pelo arquivo excel, que será enviado por um programa terminal (exemplo: Putty).

A interface controla o NumLock, da mesma forma que um PC, e os Leds de CapsLock e Scroll Lock são coletados do hardware do MSX hospedeiro, sendo que o Scroll Lock é mapeado para Katakana (ou Cirílico ou Korean).

O MSX informa a coluna (Y_Scan) e a interface responde com a linha solicitada. Se não há alterações no Y_Scan, a interface atualizará (somente) a linha apontada pelo Y_Scan, conforme os eventos do teclado PS/2 mapeados no Database.

Os eventos válidos do teclado PS/2 para esta interface são o pressionar teclas e o liberar das mesmas. Auto repeat não têm sentido para esta aplicação, por isso, é enviado ao teclado um comando para deixar o autorepeat para 2CPS e o autorepeat delay em 1,0s.

Assunto do momento

Considerações e características de contorno dos ambientes

Escolha do MCU

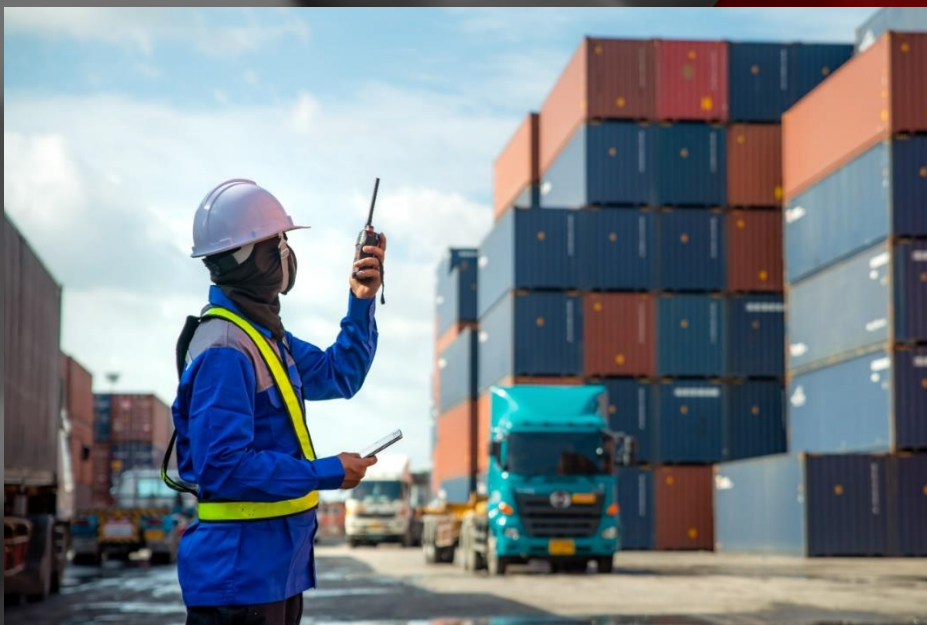
Setup do ambiente de desenvolvimento

A eletrônica

Facilidades do software

Descrição técnica do firmware

Descrição Técnica do Firmware



A técnica implementada foi a programação em *Bare metal*, ou seja, não foi implementado sobre RTOS;

A linguagem adotada foi o C++, com alguns módulos em C

A alocação das tarefas (função do Despachante) foi baseada em interrupções:

- Hardware: MSX Y-Scan & PS/2 keyboard;
- Software: System timer, com taxa de 30Hz & Timer2, com resolução de μs .

* O código fonte pode ser acessado através da minha página do github:

<https://github.com/evandrosouza-developer/PS2-MSX-STM32F1>

Descrição Técnica do Firmware

Arquivos:

- **ps2-msx.cpp**
- **Contém o main.**

Gerencia o constructor (setup) de cada módulo. O loop principal apenas atualiza os Leds do teclado PS/2.

Interface

Interface

- **flash_intelhex.c**
- **flash_intelhex.h**

Lê o arquivo Database no formato Intel Hex pela serial e atualiza a flash interna do STM32

Interface

- **msxmap.cpp**
- **msxmap.h**

Grupo de controle do mapeamento e adaptação dos teclados PS/2 e MSX. Contém as rotinas de pré-montagem do dado enviado ao MSX na interrupção ou na ausência desta. É onde fica o Database default.

Interface

- **ps2handl.cpp**
- **ps2handl.h**

Trata bit a bit a comunicação, Recebe e envia dados do teclado por byte e envia evento multibyte (o Scan Code)

Interface

- **port-def.h**

Definições de uso de Portas
Extremamente conectado ao hardware

Interface

- **hr_timer.cpp**
- **hr_timer.h**

Tem as funções de dar os delays de μs e, na função *Capture*, informar a interrupção do clock PS/2

Interface

- **sys_timer.cpp**
- **sys_timer.h**

Clock ARM, usado em 30Hz. É onde são lidos os estados de Caps e Kana. Marca a cadência do LED de status e despacha para msxmap os Scan Codes preparados

Interface

- **serial.c**
- **serial.h**

Implementa todas as entradas e saídas com administração de buffer e Xon/Xoff protocol

Cada .cpp ou .c contém as rotinas de setup, rotinas de trabalho e suas respectivas Interrupt Service Routines.

Os .h são usados para definições e divulgar as rotinas públicas, além da adaptação de header para CPP.

THANKS

Solution