



**ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO**
Departamento de Engenharia de Sistemas Eletrônicos PSI -
EPUSP



PSI 2618
CIRCUITOS ELETRÔNICOS AUTOMOTIVOS

Gerenciamento de Motores
ECU

Prof. Armando Laganá

Abril 2017

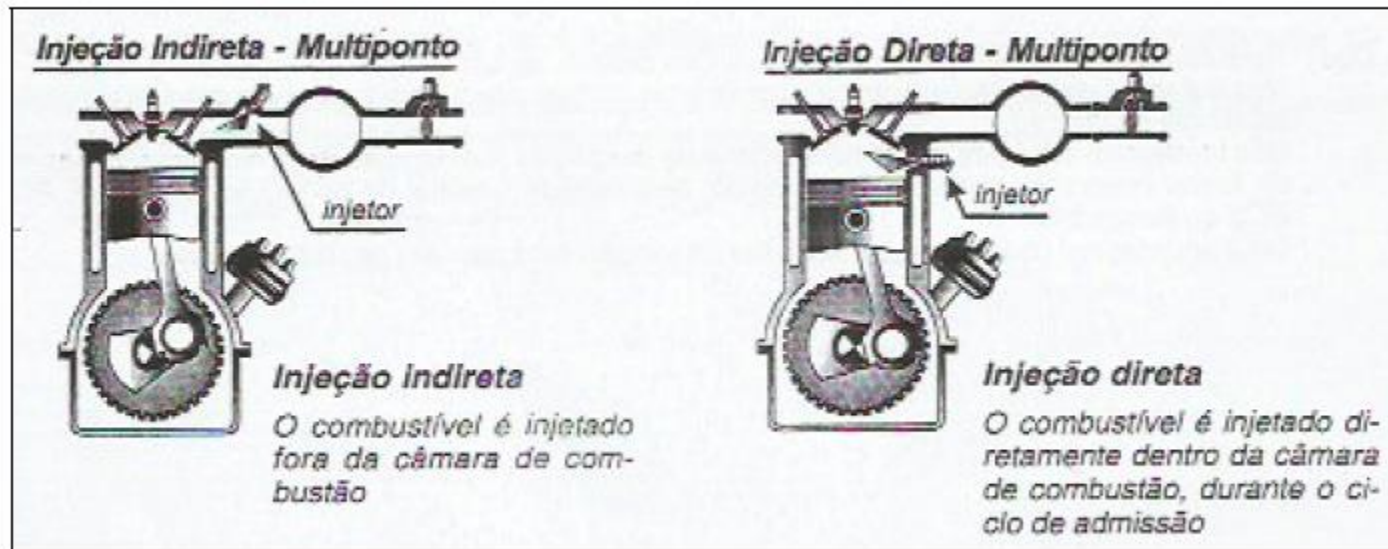
- 1.ELEMENTOS
- 2.HARDWARE
- 3.FIRMWARE
- 4.RESULTADOS

Figura 2 – Controle utilizado pelos pesquisadores



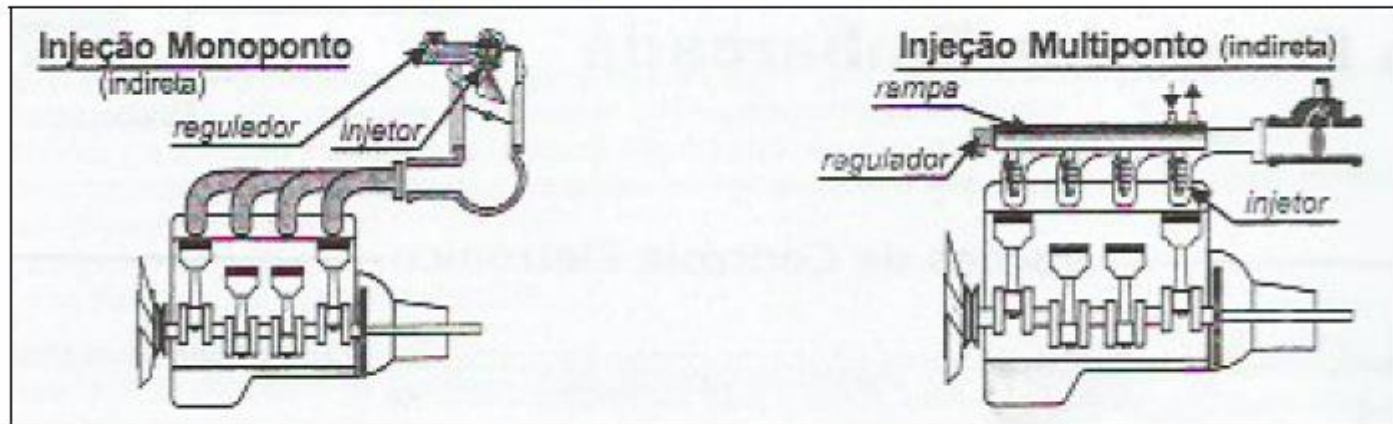
Fonte: BBC News (2013)

Figura 3 – Injeção Direta e Indireta



Fonte: Manavella (1996)

Figura 4 – Injeção Monoponto e Multiponto



Fonte: [Manavella \(1996\)](#)

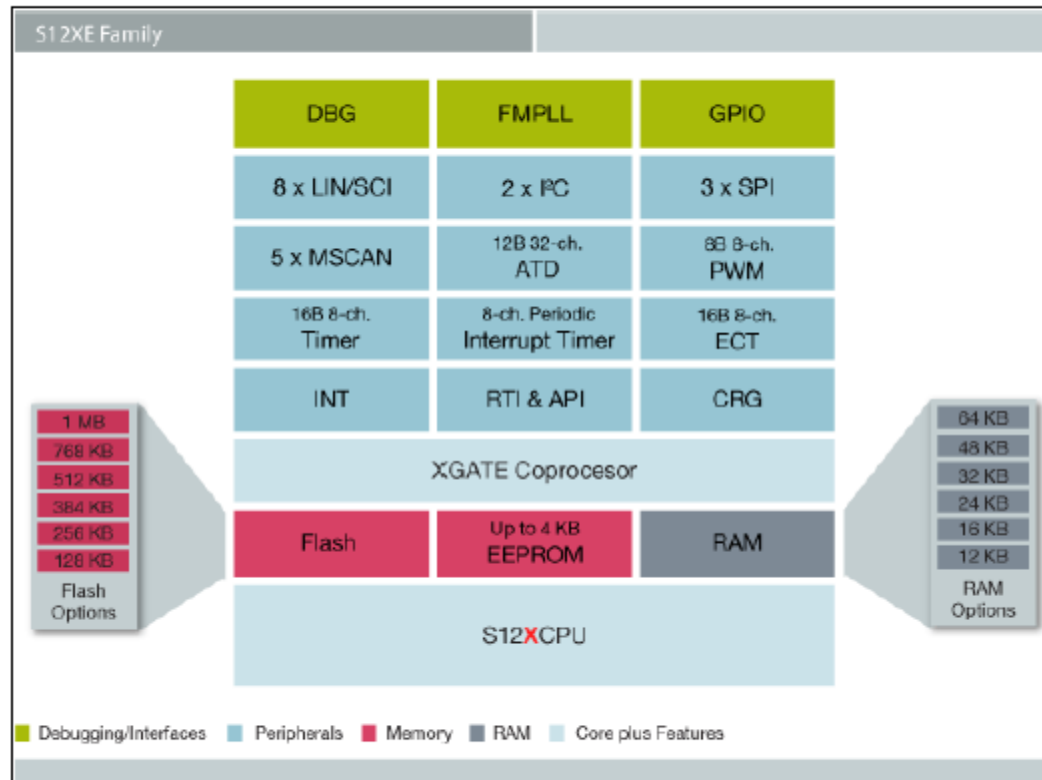
MICROCONTROLADOR

Figura 5 – O microcontrolador automotivo S12XE da Freescale



Fonte: o autor

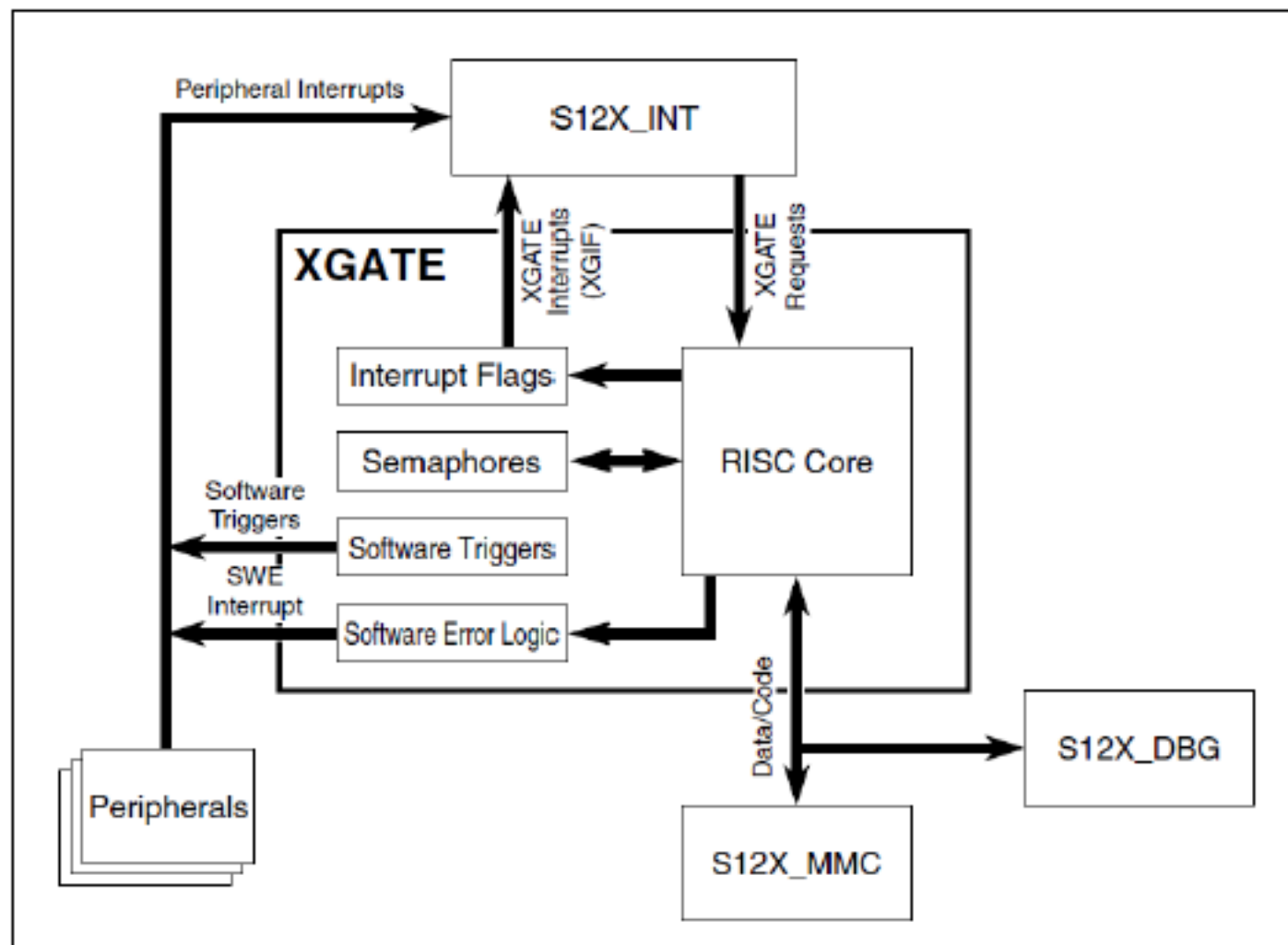
Figura 6 – Diagrama de blocos da família de microcontroladores S12XE



Fonte: Freescale (2013)

- 256 kB de memória flash e 16 kB de memória RAM;
- 16 canais ADC, com resolução de 12 bits e tempo de conversão de até 3 microssegundos;
- 4 *Timers* de 16 bits com canais e interrupções independentes;
- 8 canais que, associados a um *timer* de 16 bits, possuem recursos avançados como captura de entrada e comparação de saída;
- 3 interfaces CAN, 4 interfaces seriais assíncronas (UART), 3 interfaces SPI e 1 interface I2C;
- 8 canais para geração de PWM de 8 bits

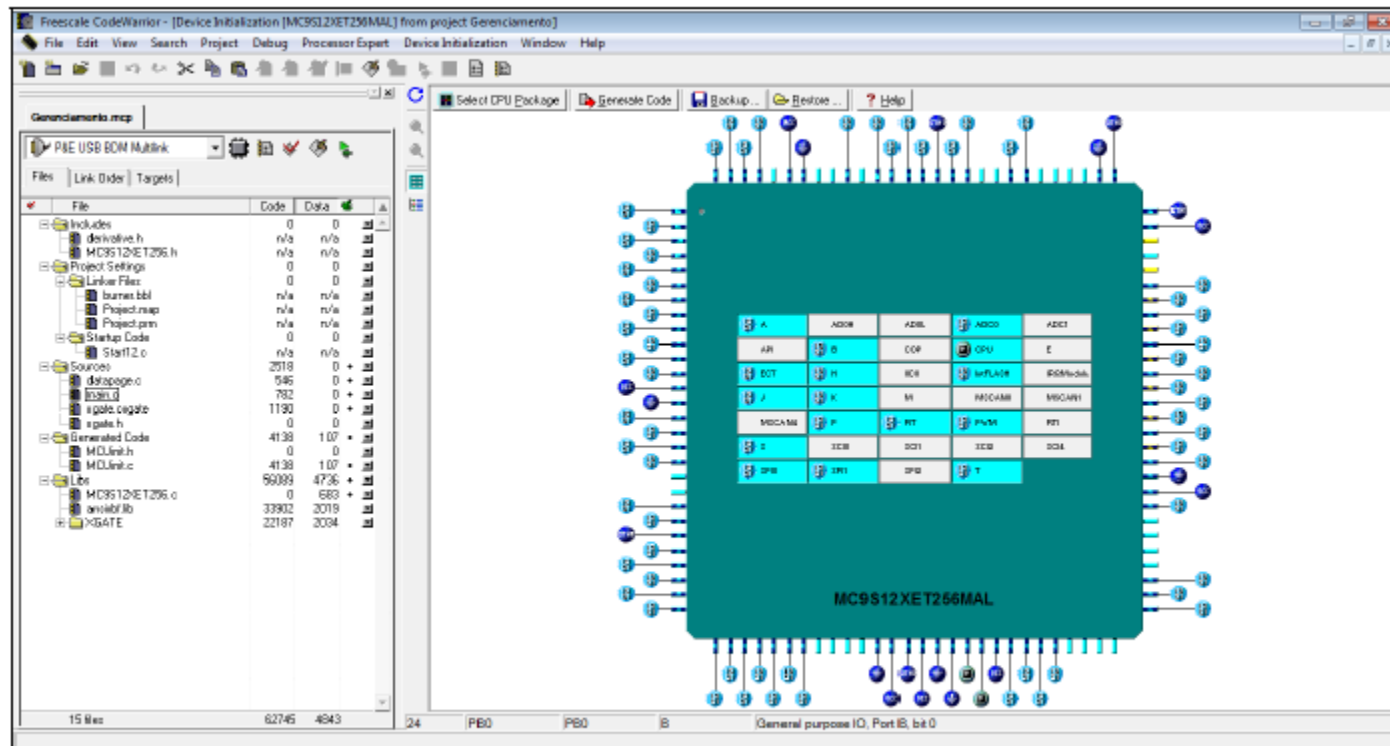
Figura 7 – Diagrama de blocos do coprocessador XGate



Fonte: Freescale (2012)

A ferramenta Codewarrior v5.1 HCS12 da Freescale

Figura 8 – A ferramenta Codewarrior v5.1 da Freescale



Fonte: o autor

Figura 9 – Exemplo de configuração do periférico SPI na interface *Device Initialization*

The image shows a software configuration window titled "Inspector Init_SPI". It is divided into two main sections: "Component Parameters" on the left and "Register Details" on the right. The "Component Parameters" section is expanded to show "Clock settings" and "Pins" configurations. The "Register Details" section shows a table of SPI registers with their addresses, initial values, and bit maps.

Component Parameters		Register Details			
	Value	Name	Address	Init. value	Register Map
Device	SPI0	SPI0CR1	0x00D8	50	H
Settings		SPI0CR2	0x00D9	40	H
Clock settings		SPI0BR	0x00DA	01	H
Value of Pselection	0	SPI0SR	0x00DB	20	H
Value of Selection	1	SPI0RR	0x00DC	00	H
Frequency	12.5 MHz				
Mode Select	Master				
Clock Polarity	active-high				
Clock Phase	First edge				
Data shift order	MSB first				
Bidirectional Mode	Disabled				
Output enable in Bidirect.	no				
SS pin function	General Purpose I/O				
Stop in Wait Mode	no				
Data Width	16 bits				
Pins					
SCK pin					
SCK pin	PM5_TxCAN0_TxCAN4_SCK0				
SCK pull resistor	no initialization				
SCK pin open drain	push-pull				
Reduced drive for PM5	no				
MISO pin allocation	Enabled				
MISO pin	PM2_RxCAN1_RxCAN0_MISO				
MISO pull resistor	no initialization				
MISO pin open drain	push-pull				
Reduced drive for PM2	no				
MOSI pin allocation	Enabled				
MOSI pin	PM4_RxCAN0_RxCAN4_MOSI				
MOSI pull resistor	no initialization				

The "Register Details" table shows the following registers:

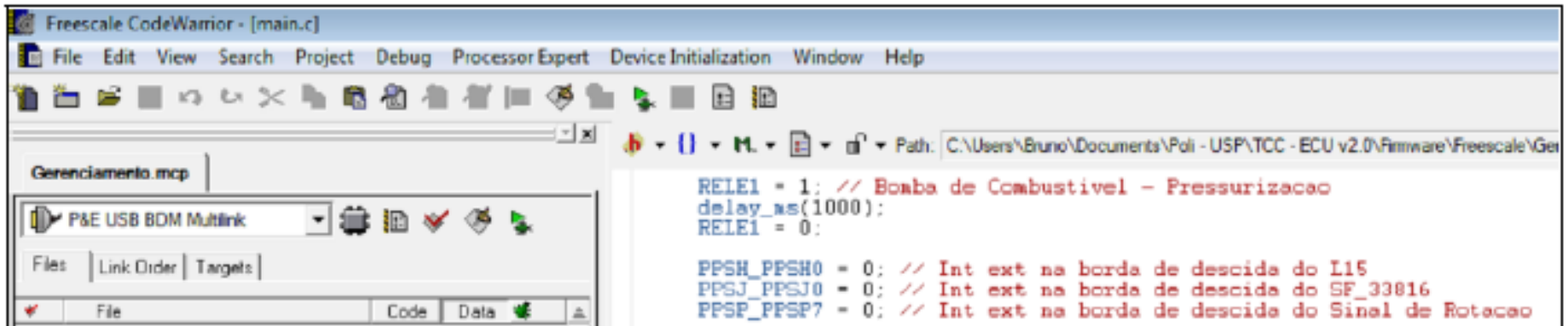
Name	Address	Init. value	Register Map
SPI0CR1	0x00D8	50	H
SPI0CR2	0x00D9	40	H
SPI0BR	0x00DA	01	H
SPI0SR	0x00DB	20	H
SPI0RR	0x00DC	00	H

A tooltip is visible over the SPI0BR register, stating: "SPI Baud Rate selection value used for setting SPI communication frequency. This item modifies the SPR0.SPR1 and SPR2 bits in the SPIBR register."

At the bottom of the window, there are several control elements: a checkbox for "View Register Map" (checked), a legend for the Register Map (1, 0, undefined/reserved/read-only), and buttons for "Disable Peripheral Initialization", "OK", "Cancel", and a help icon.

Fonte: o autor

Figura 10 – Trecho de código retirado do uC de Gerenciamento



The image shows a screenshot of the Freescale CodeWarrior IDE. The window title is "Freescale CodeWarrior - [main.c]". The menu bar includes "File", "Edit", "View", "Search", "Project", "Debug", "Processor Expert", "Device Initialization", "Window", and "Help". The toolbar contains various icons for file operations and development. The project name is "Gerenciamento.mcp". The path is "C:\Users\Bruno\Documents\Poli - USP\TCC - ECU v2.0\Firmware\Freescale\Ger". The code in the editor is as follows:

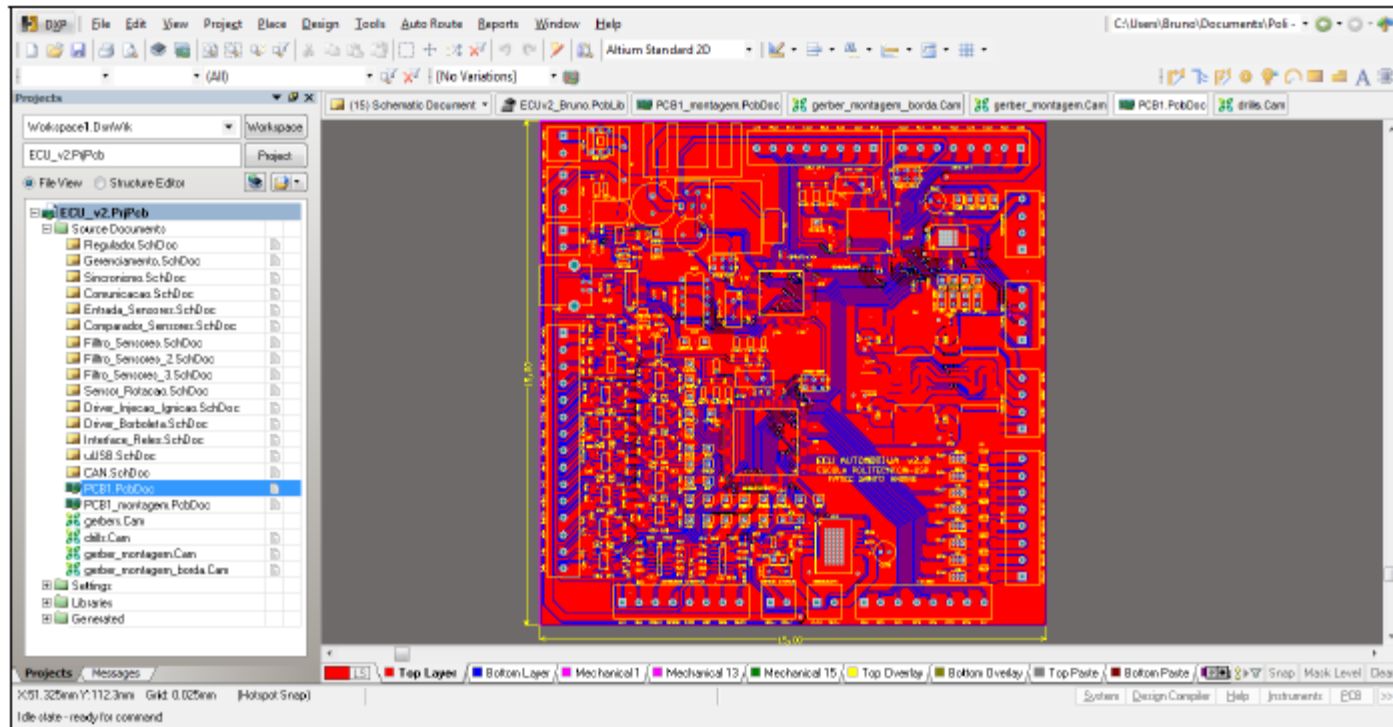
```
RELE1 = 1; // Bomba de Combustivel - Pressurizacao
delay_ms(1000);
RELE1 = 0;

PPSH_PPSH0 = 0; // Int ext na borda de descida do I15
PPSJ_PPSJ0 = 0; // Int ext na borda de descida do SF_33016
PPSP_PPSP7 = 0; // Int ext na borda de descida do Sinal de Rotacao
```

Fonte: o autor

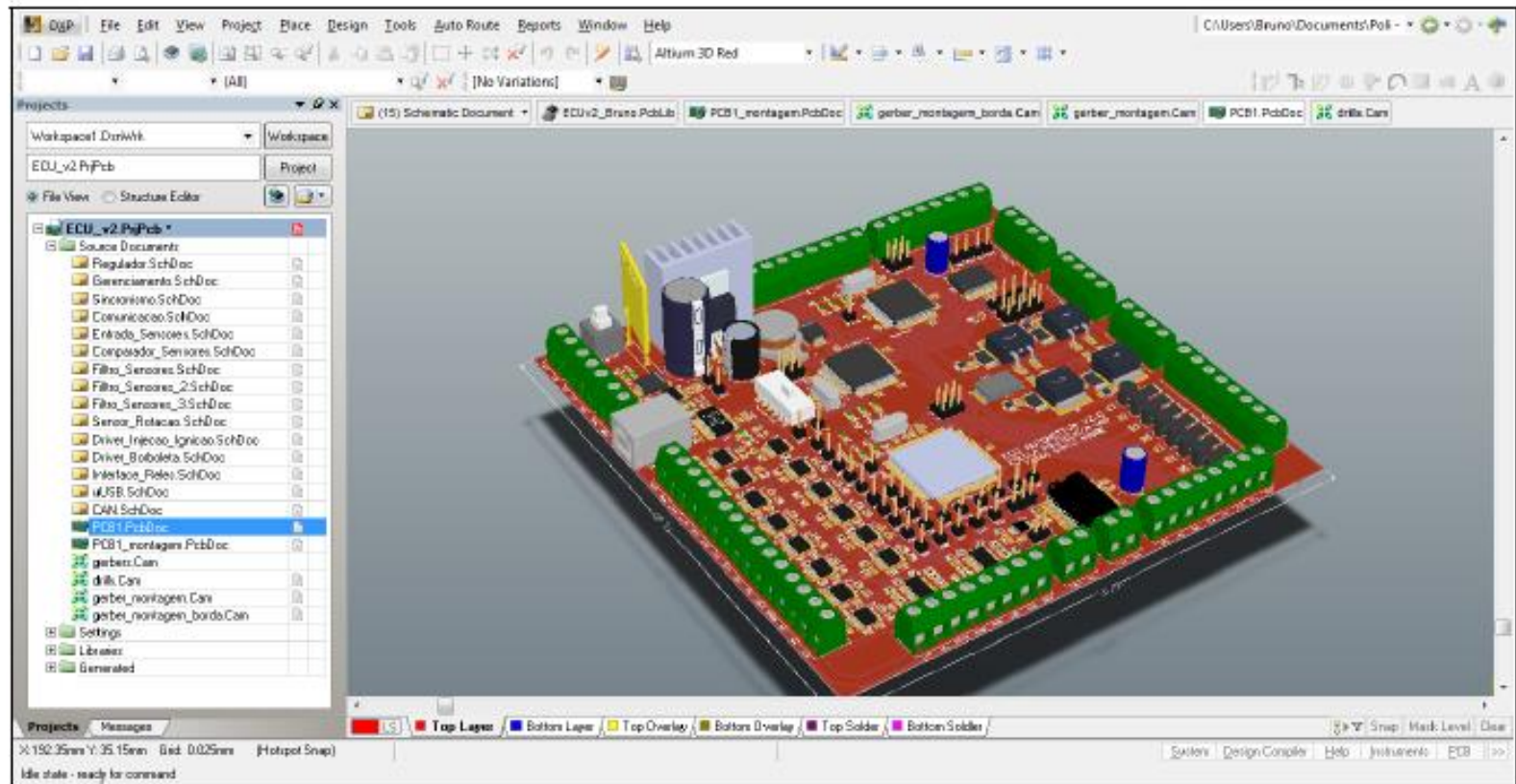
2.3.6 A ferramenta Altium Designer

Figura 12 – A ferramenta Altium Designer



Fonte: o autor

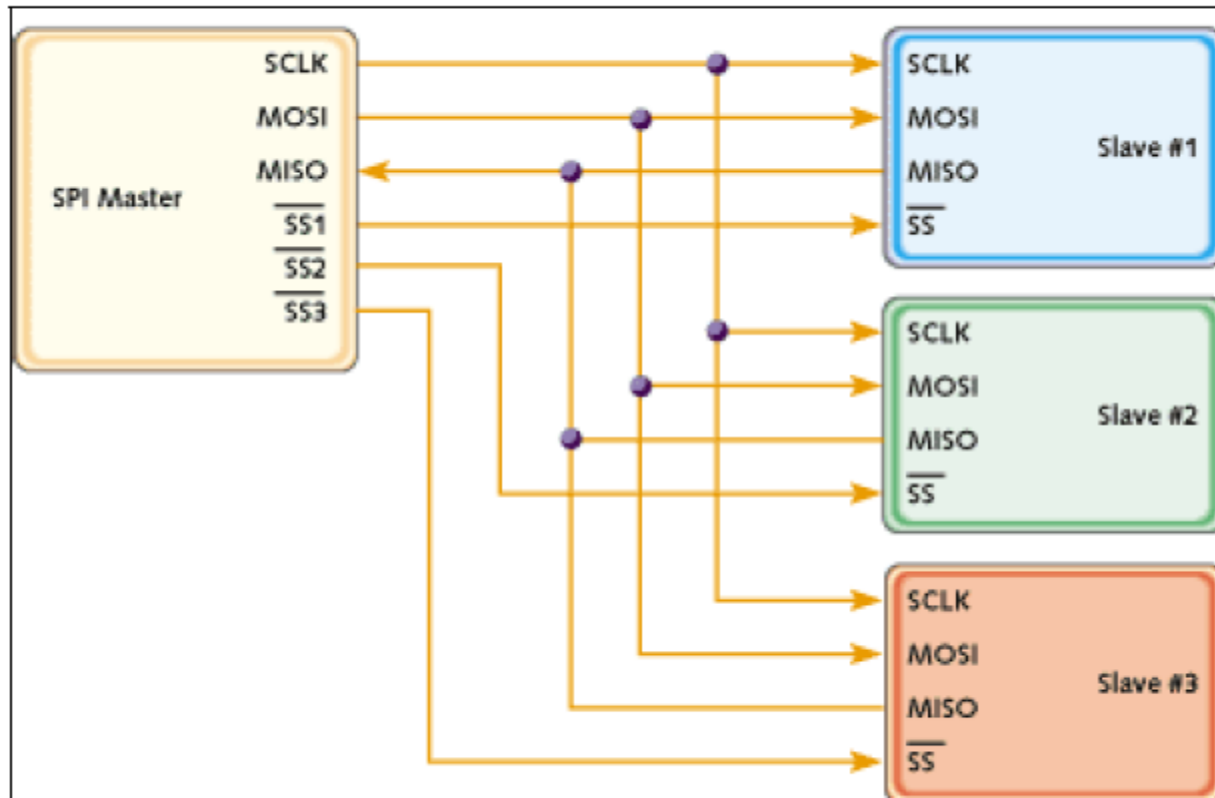
Figura 13 – Modelo 3D da ECU v2.0 projetada com a ferramenta Altium Designer



Fonte: o autor

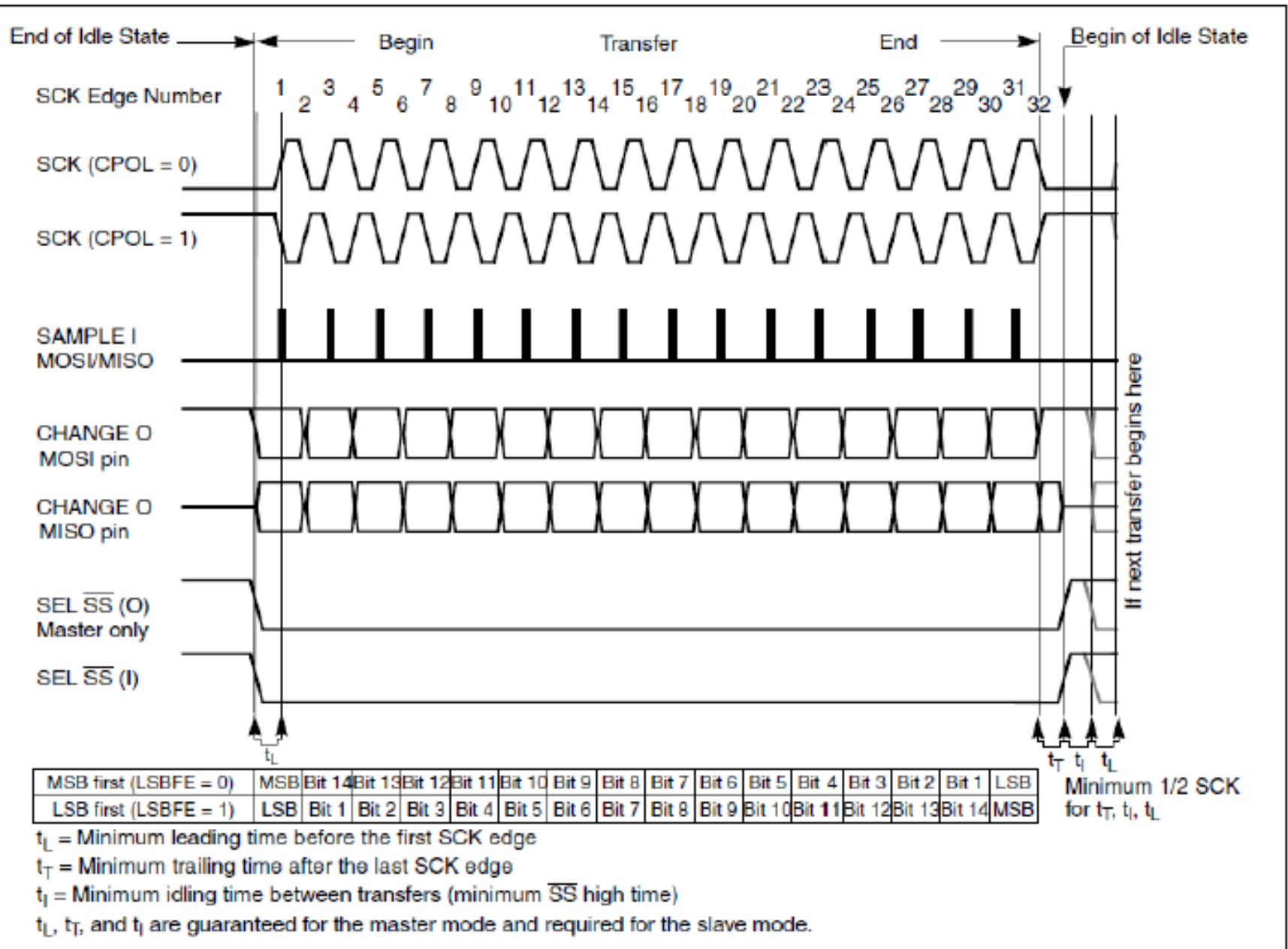
SPI

Figura 14 – Esquema de ligações de uma rede de comunicação SPI



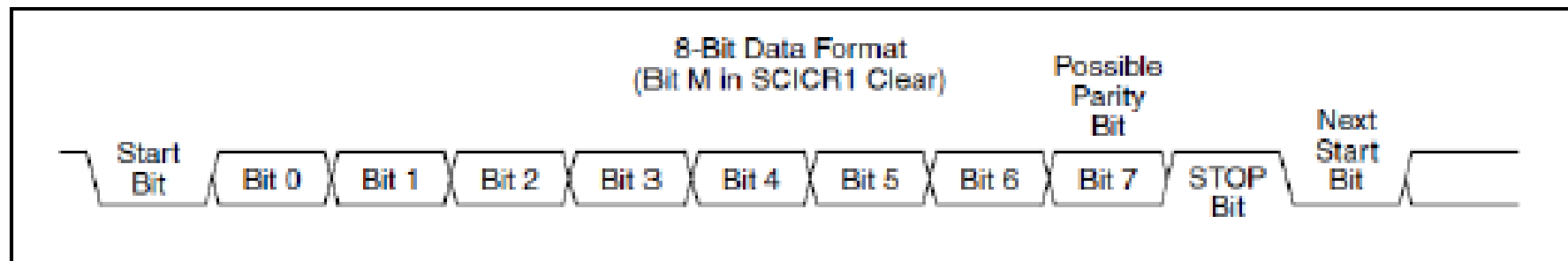
Fonte: Kalinsky e Kalinsky (2002)

Figura 15 – Carta de tempos da comunicação SPI no S12X



O protocolo UART

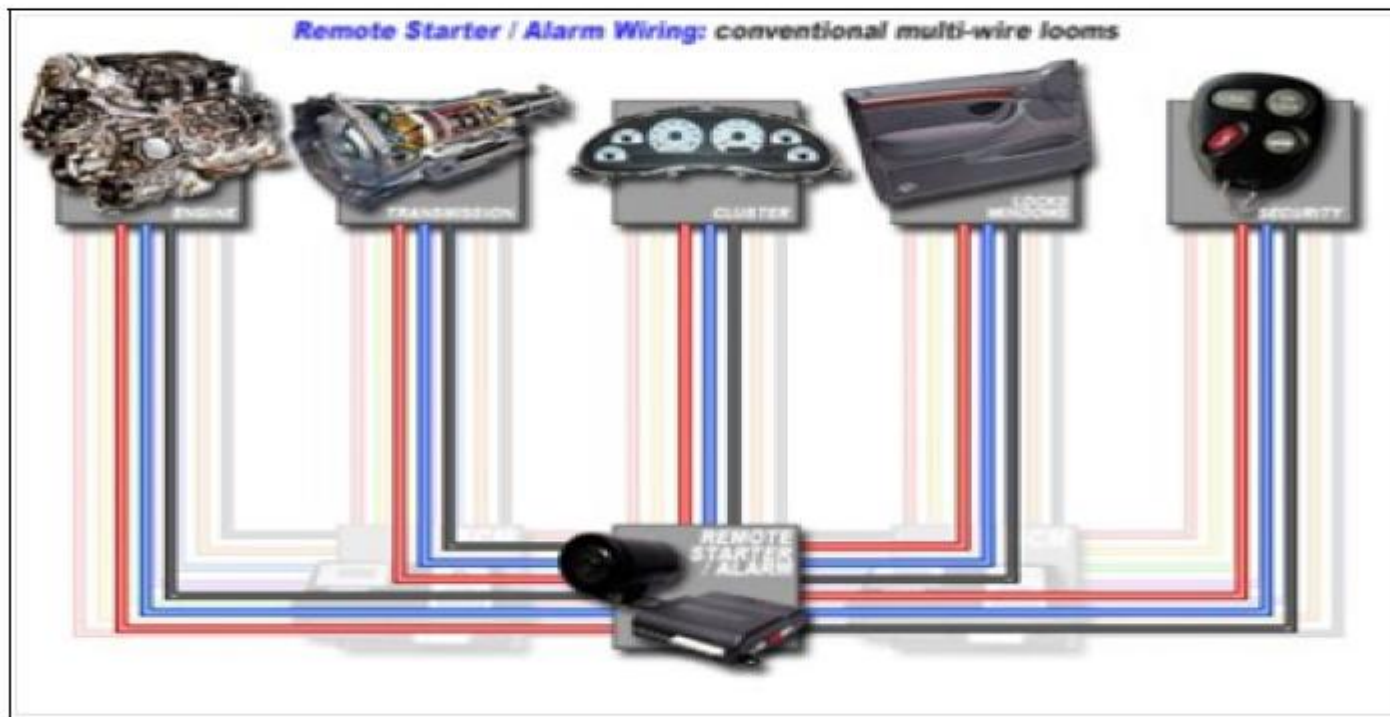
Figura 16 – Formato do protocolo de comunicação UART



Fonte: Freescale (2012)

2.3.7.3 O protocolo CAN

Figura 17 – Exemplo de CAN BUS

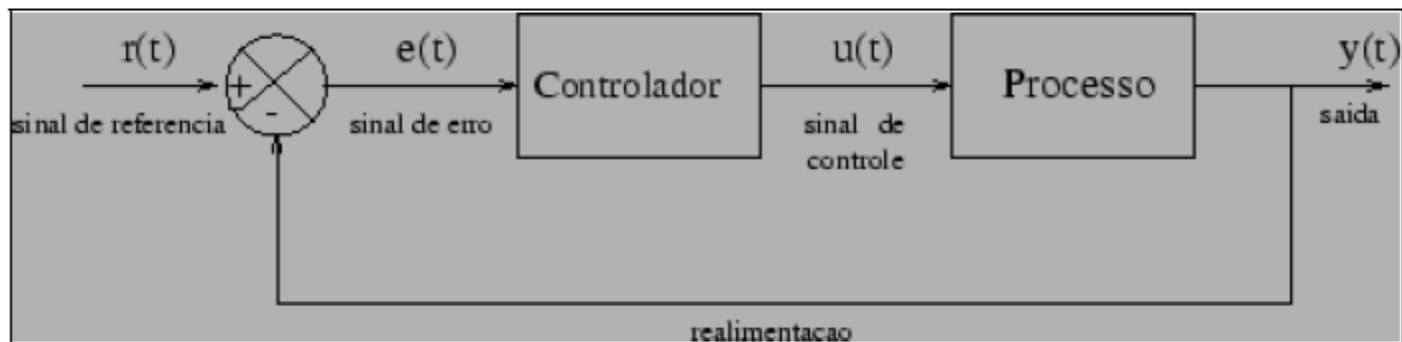


Fonte: SCARPINETTI e SOARES (2012)

MALHA DE CONTROLE

2.3.8 Controladores do tipo proporcional-mais-integral (PI)

Figura 18 – Controle em malha fechada



Fonte: Silva e Bazanella (2000)

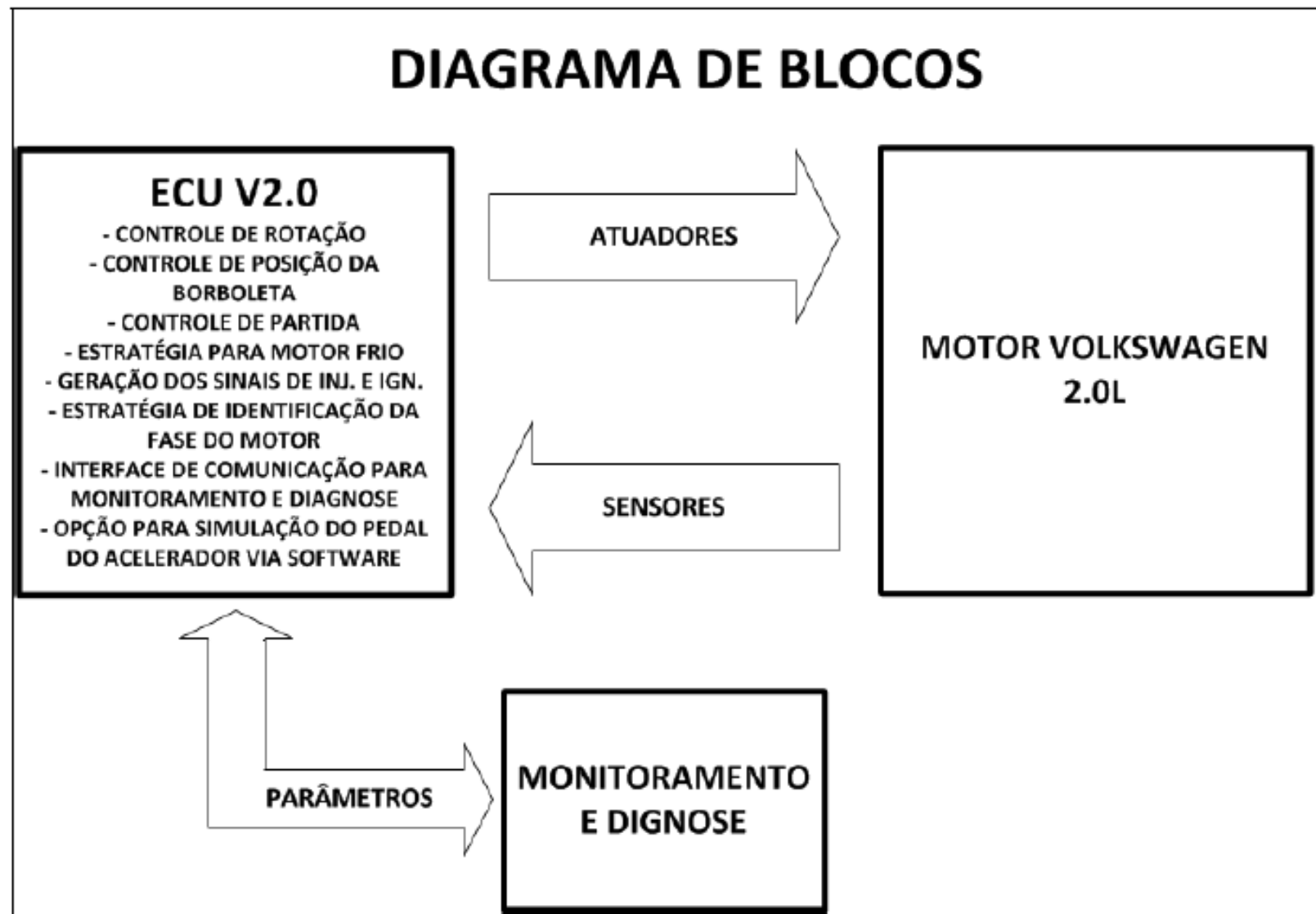
$$u(t) = K_p \times e(t) + K_i \times \int e(t)dt$$

Tabela 1 – Efeitos ao se alterar as constantes do PI (Adaptado de Ang, Chong e Li (2005))

Constante Incrementada	Tempo de Subida	de Sobressinal	Tempo de Estabilização	Erro Estacionário	Estabilidade
K_p	Diminui	Aumenta	Aumenta Pouco	Diminui	Diminui
K_I	Aumenta Pouco	Aumenta	Aumenta	Diminui Muito	Diminui

3.1 Projeto Conceitual

Figura 19 – Diagrama de blocos do projeto conceitual



SENSORES

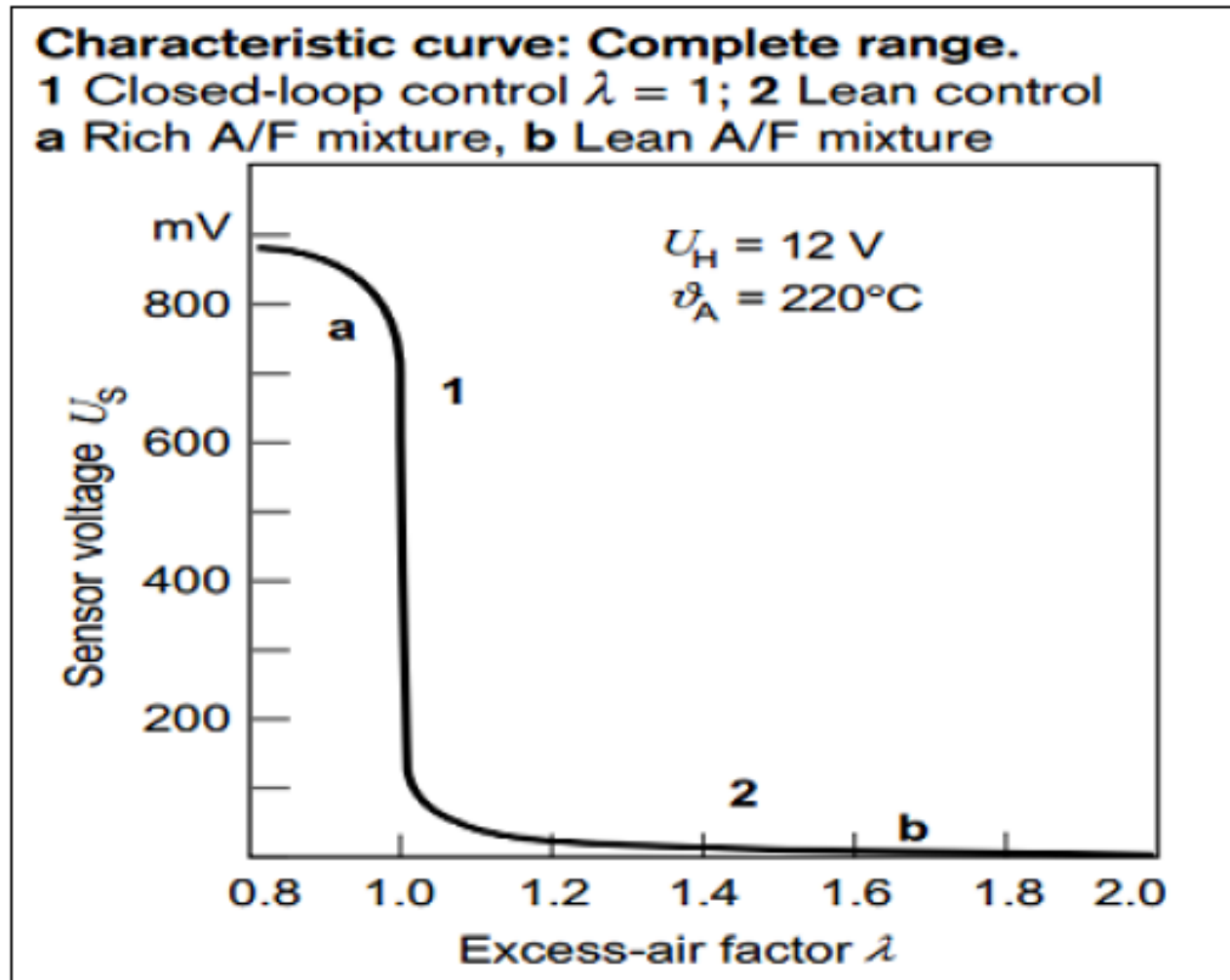
3.1.1.1 Sonda Lambda

Figura 20 – Sonda Lambda tipo LSM 11



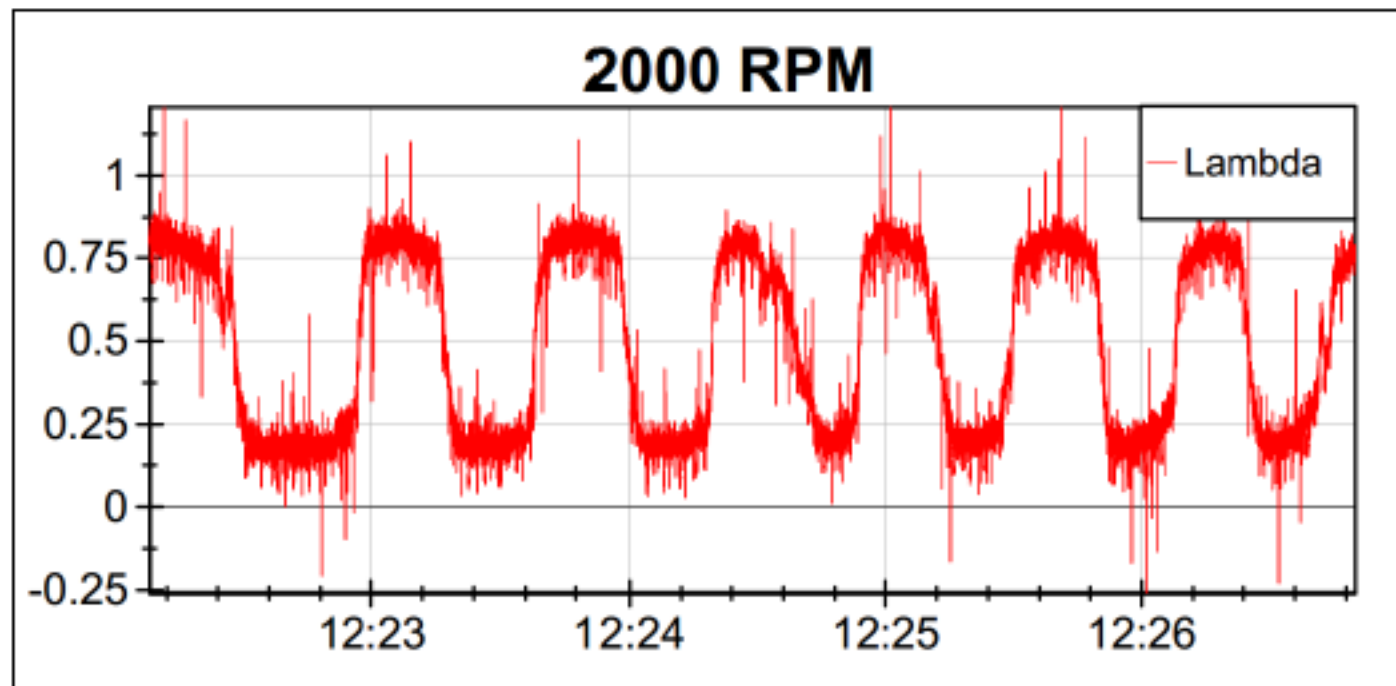
Fonte: Bosch (2013d)

Figura 21 – Curva característica da sonda lambda



Fonte: Bosch (2013d)

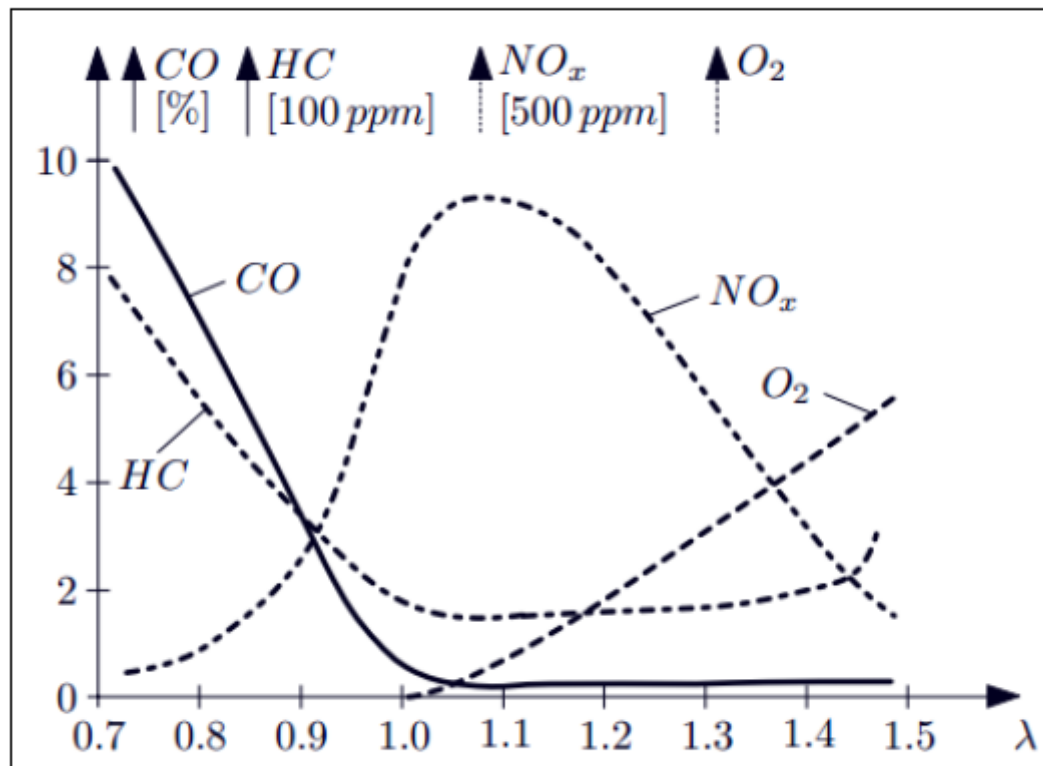
Figura 22 – Controle em malha fechada do fator lambda



Fonte: Escola Politécnica da USP; Fatec Santo André (2013)

- $\lambda < 1$: Emissão elevada de hidrocarbonetos (HC) e monóxido de carbono (CO). Também entrega alta potência devido à alta quantidade de combustível injetada;
- $\lambda = 1$: Reagentes da combustão em proporção estequiométrica, com emissões de poluentes minimizadas (principalmente com uso de um catalisador para tratar os gases de escape). Além disto, fornece uma razoável potência;
- $\lambda > 1$: Apresenta boa eficiência devido à maior massa de ar admitida pelo motor. Todavia ocorre aumento considerável da emissão de óxidos nitrosos (NOx), o que pode comprometer a vida útil do catalisador do veículo.

Figura 23 – Emissão de poluentes x fator lambda



Fonte: Kiencke e Nielsen (2005)

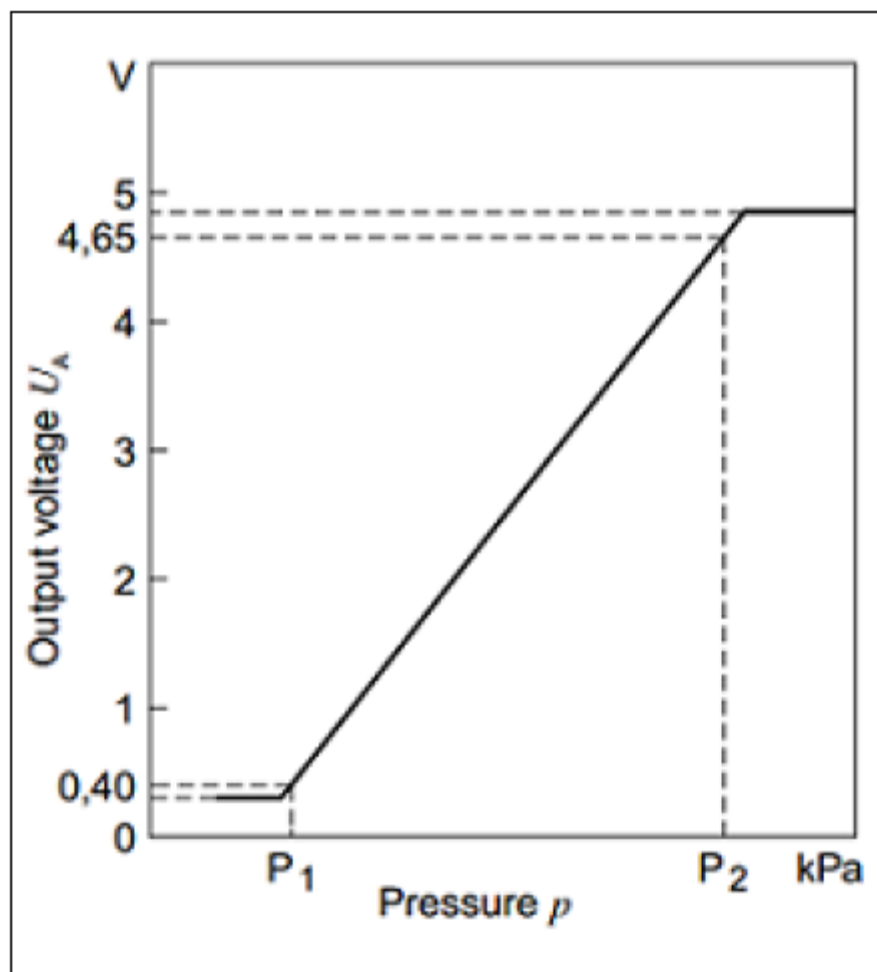
3.1.1.2 Sensor MAP

Figura 24 – Sensor MAP



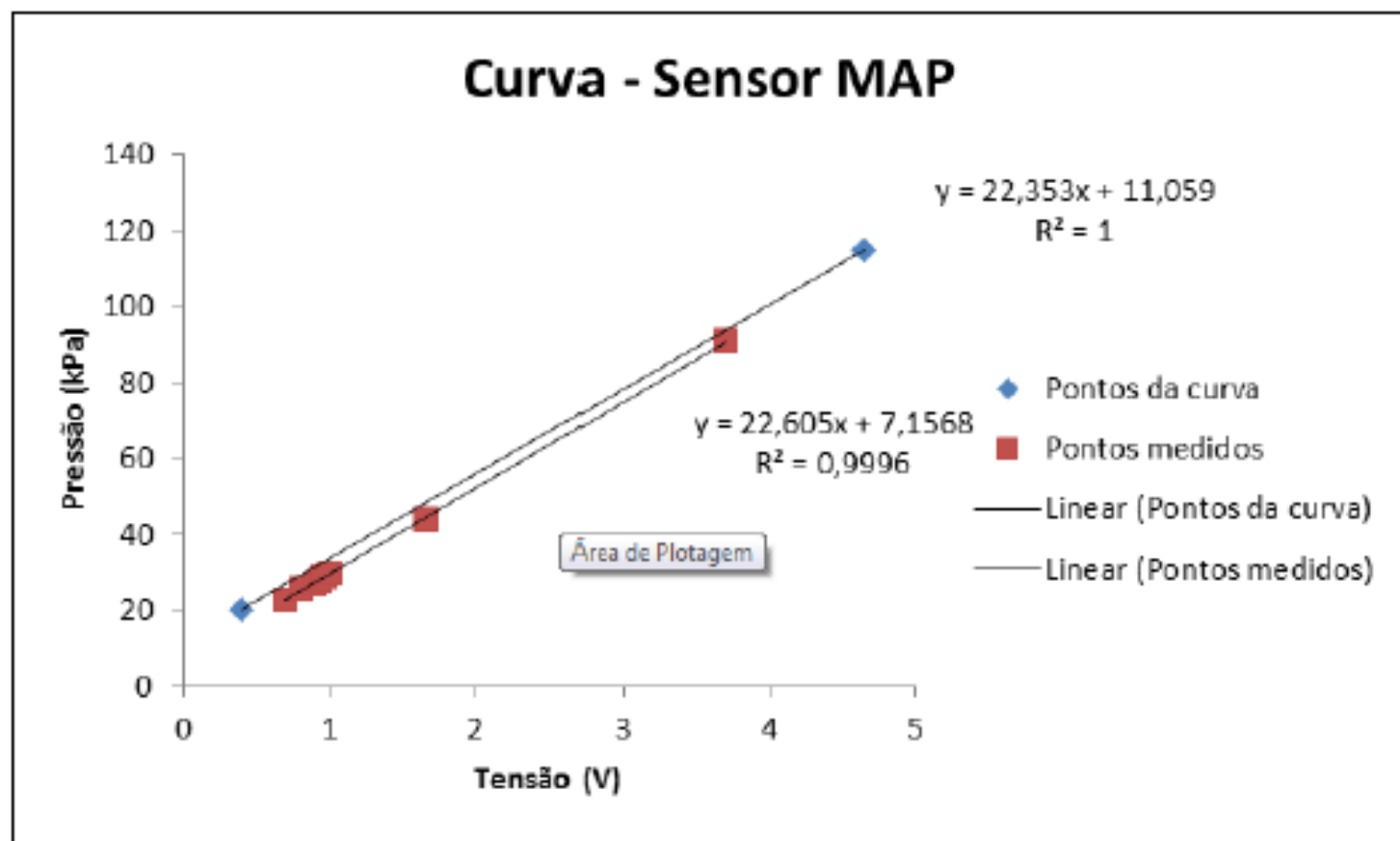
Fonte: Bosch (2013b)

Figura 25 – Curva característica do sensor MAP



Fonte: Bosch (2013b)

Figura 26 – Interpolação da curva característica do sensor MAP



Fonte: o autor

3.1.1.3 Sensor de temperatura do ar

Figura 27 – Curva característica do sensor de temperatura do ar

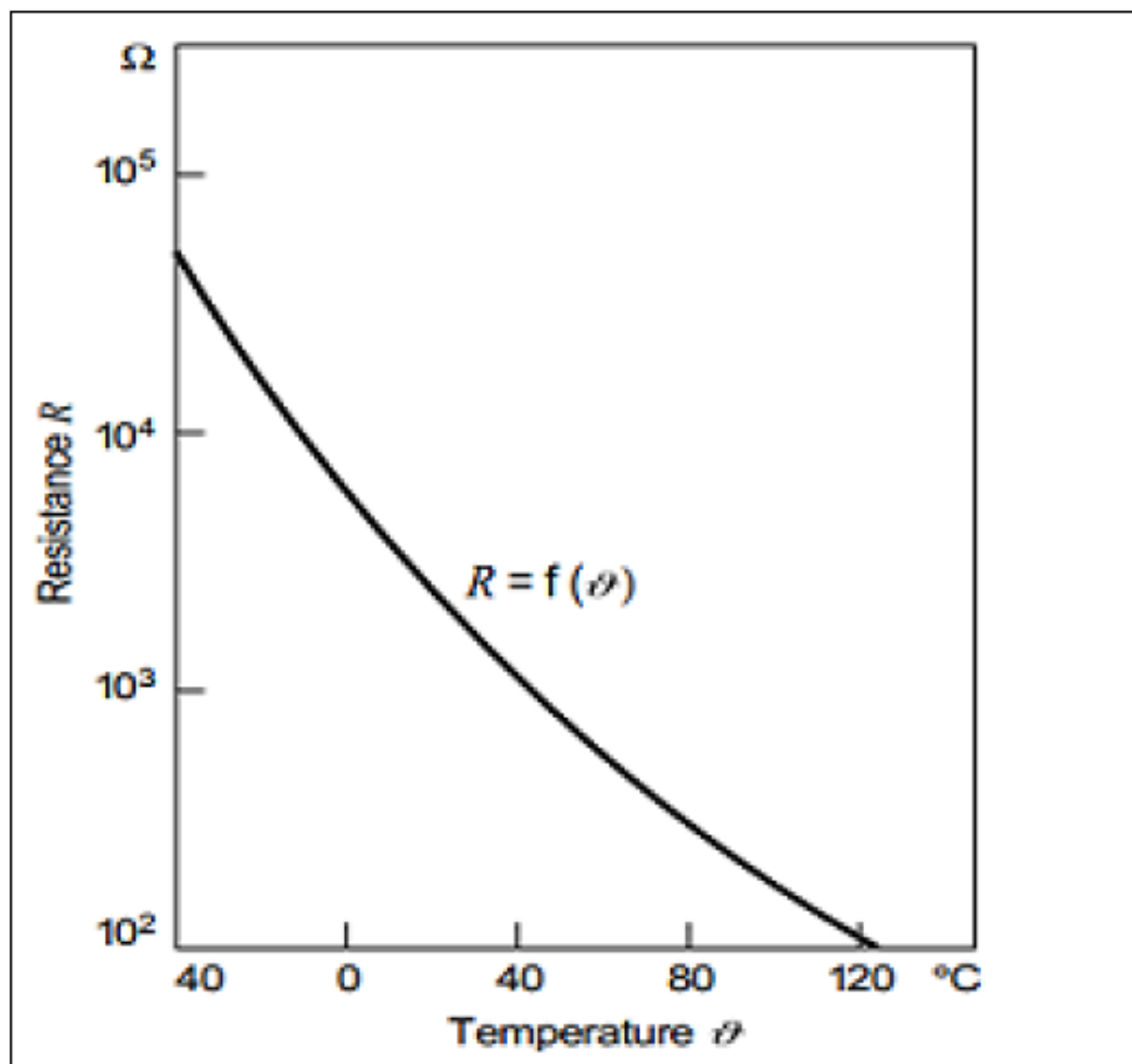
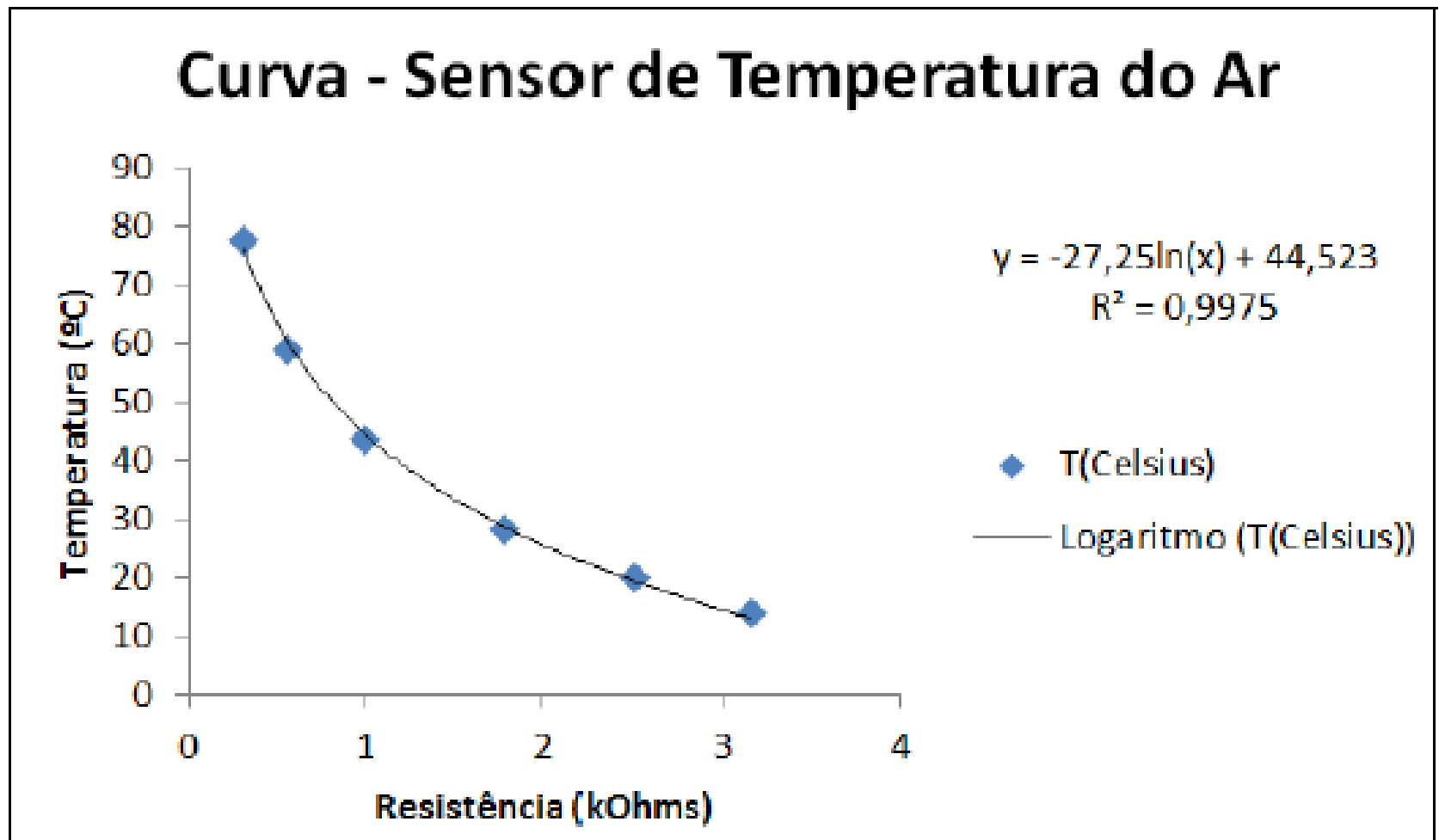
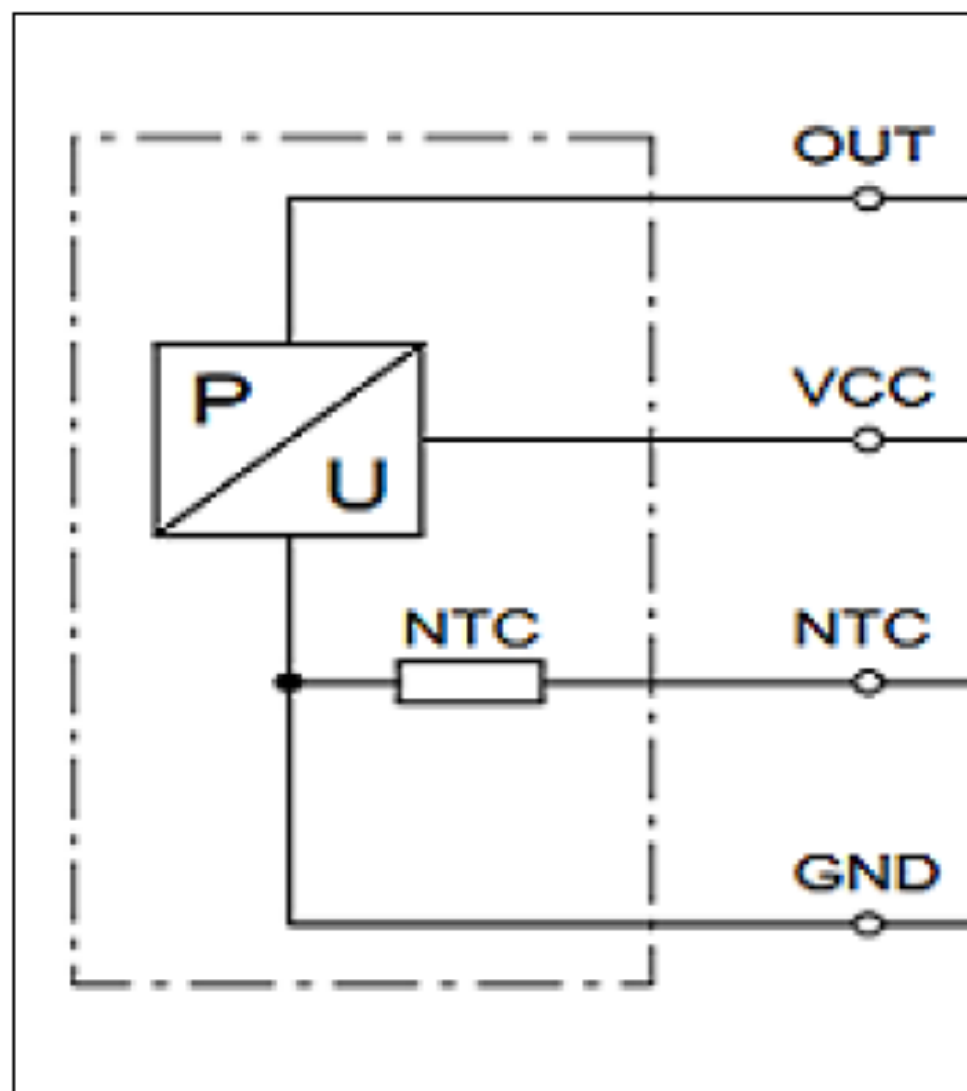


Figura 28 – Interpolação da curva característica do sensor de temperatura do ar



Fonte: o autor

Figura 29 – Diagrama do MAP com o sensor de temperatura integrado



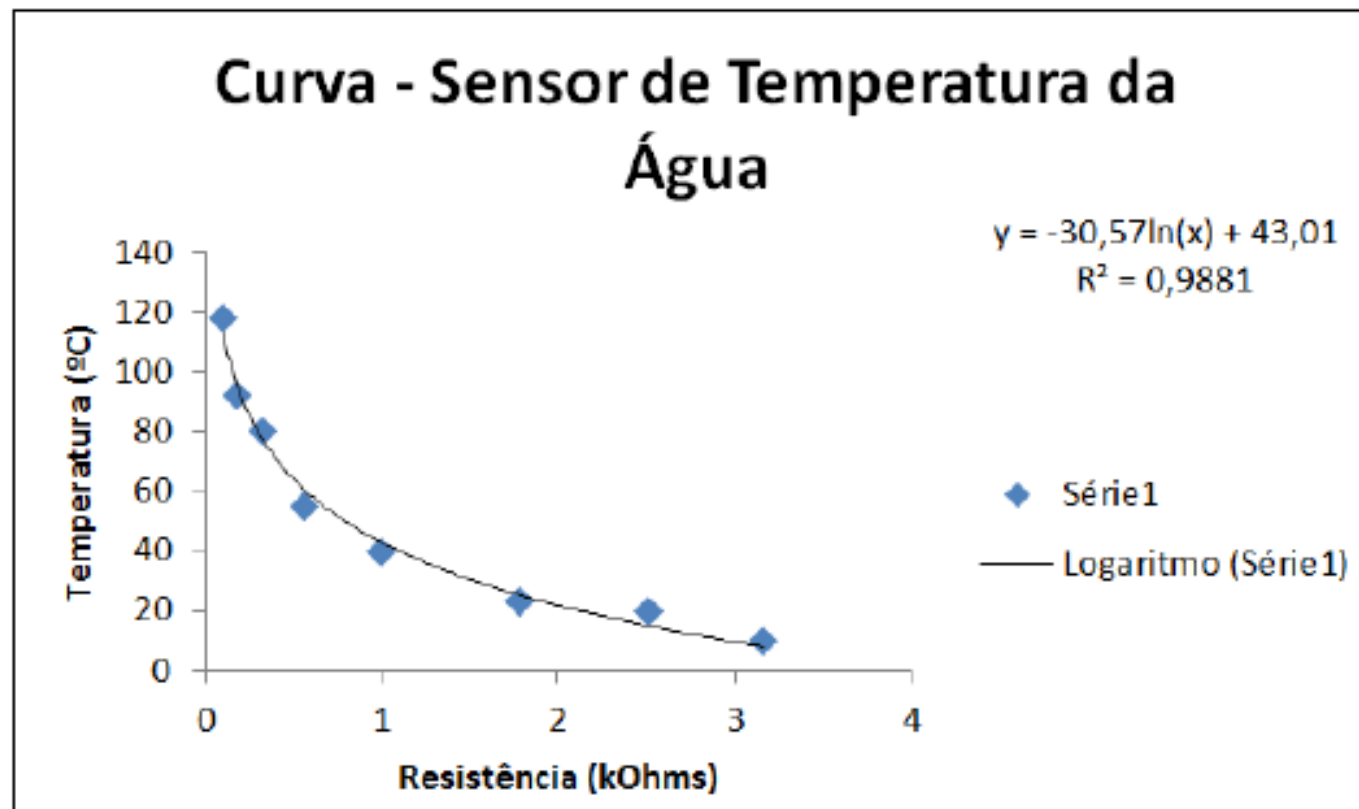
Fonte: Bosch (2013b)

Figura 30 – Sensor de temperatura da água



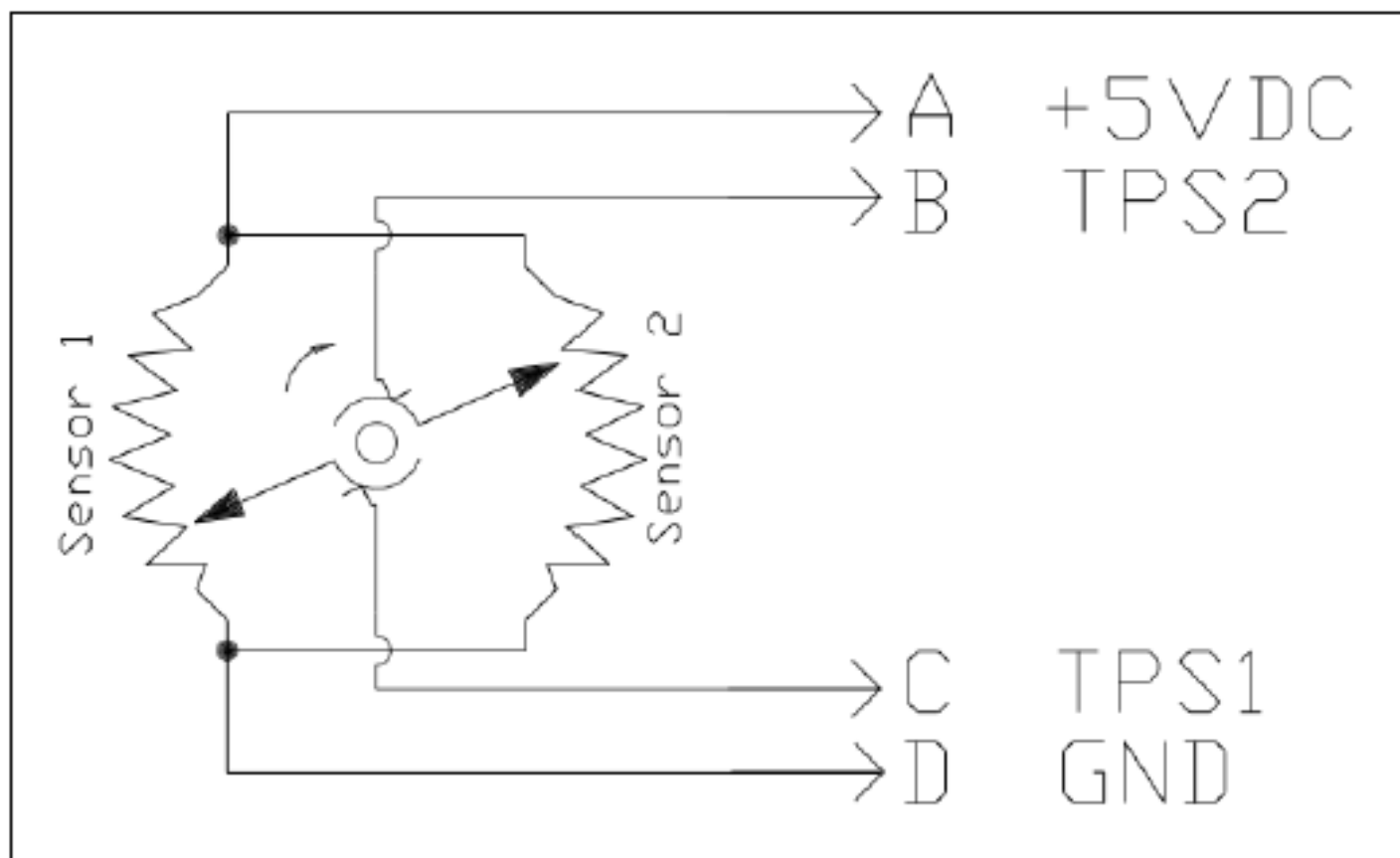
Fonte: [Bosch \(2013d\)](#)

Figura 31 – Interpolação da curva característica do sensor de temperatura da água



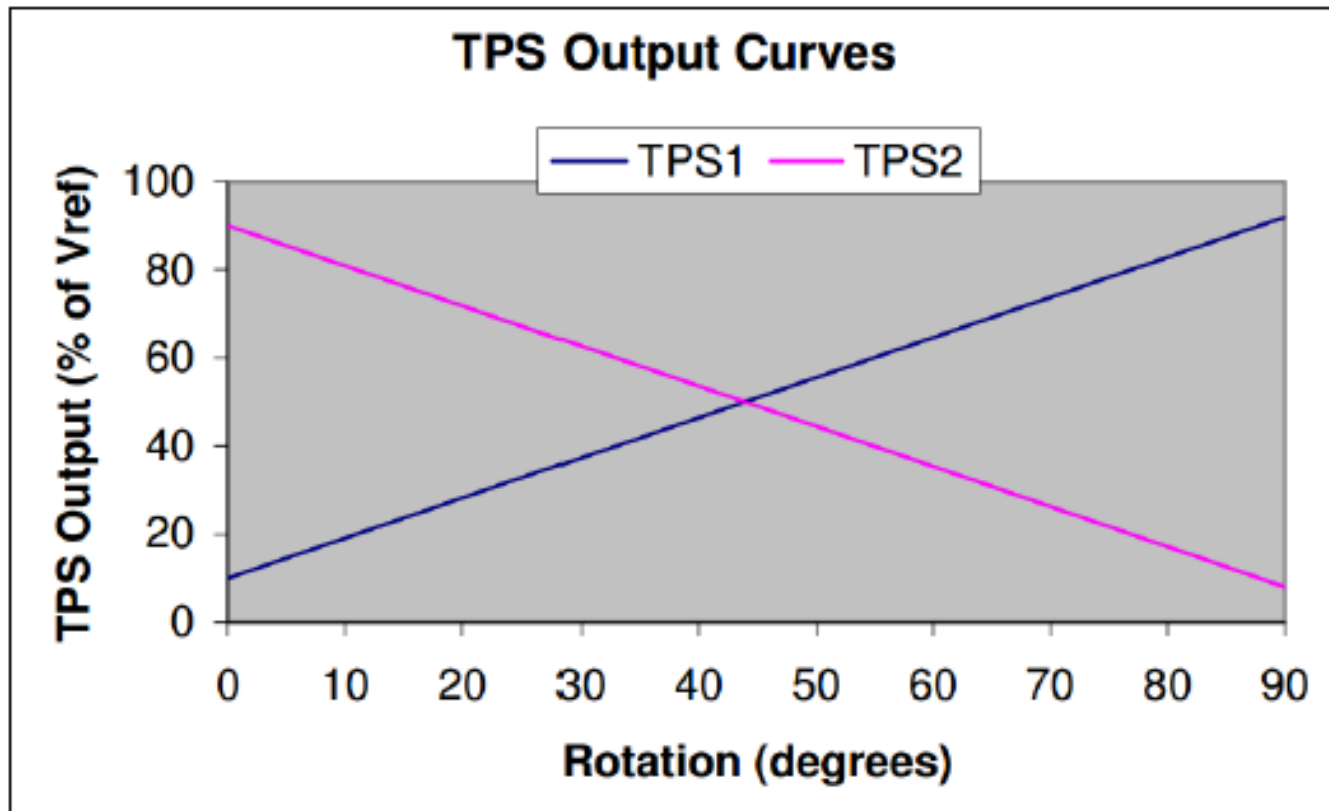
Fonte: o autor

Figura 32 – Diagrama do sensor TPS da válvula borboleta



Fonte: Delphi (2003)

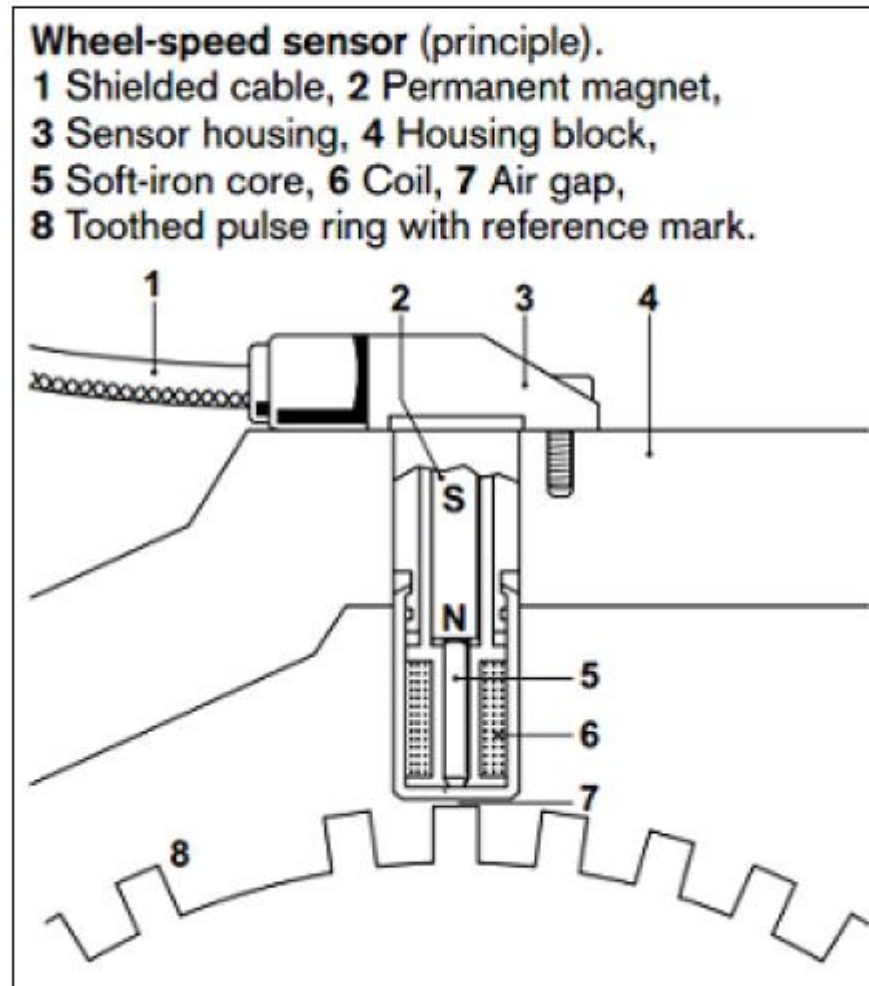
Figura 33 – Curva da variação do TPS em função da posição da VB



Fonte: Delphi (2003)

Sensor de rotação

Figura 34 – Diagrama do sensor de relutância



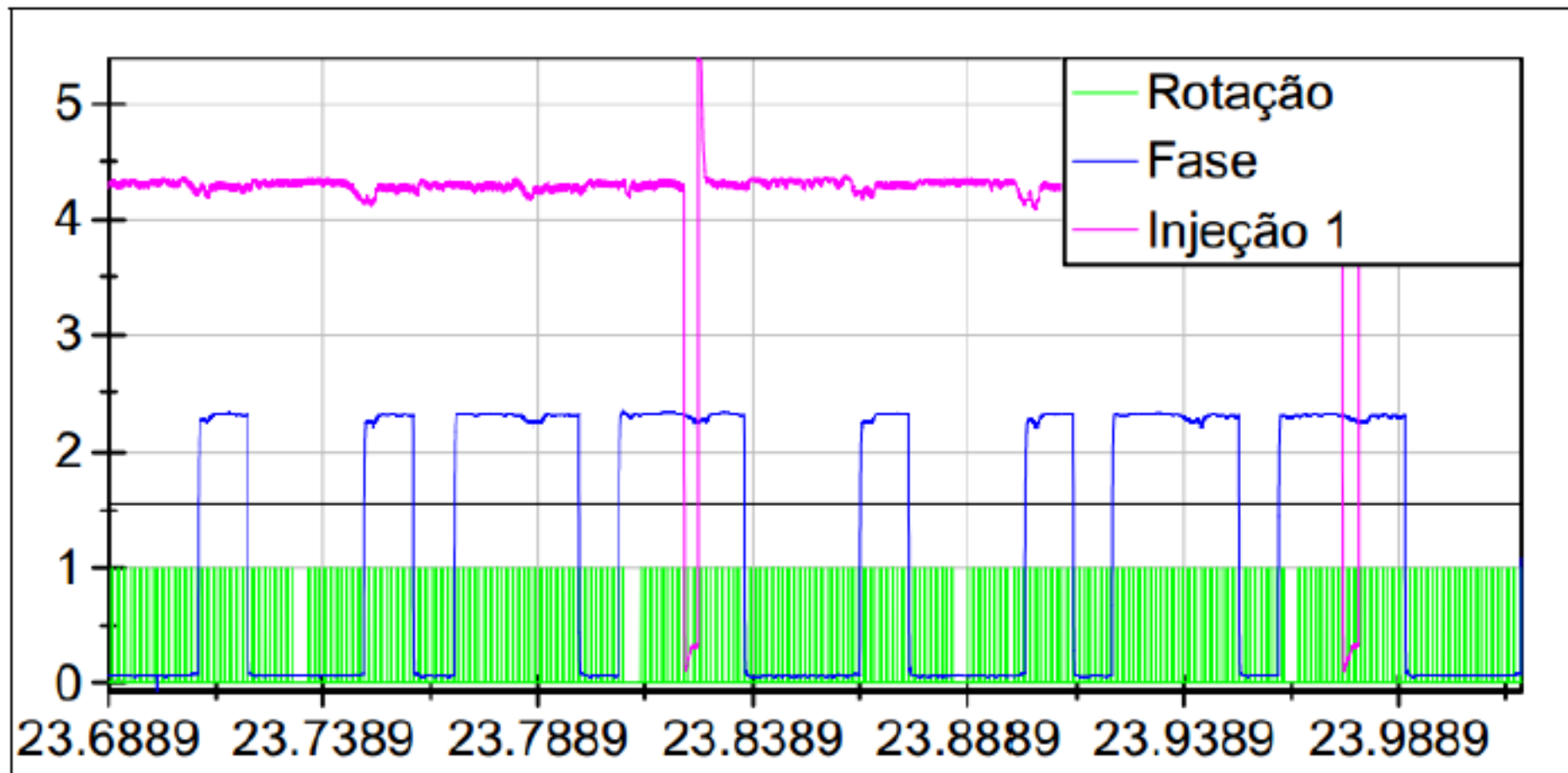
Fonte: Bosch (2013d)

Figura 35 – Roda Fônica equipada com o sensor de rotação



Fonte: Igniflex... (2013)

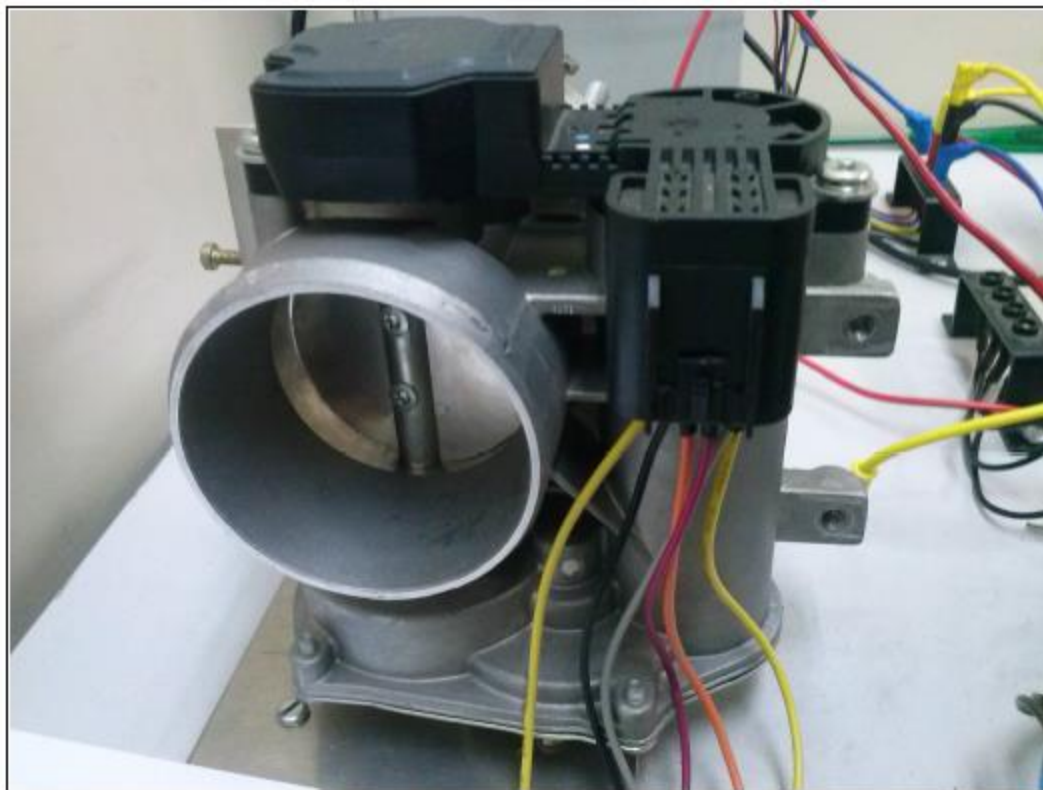
Figura 36 – Acionamento da injeção do cilindro 1 de acordo com o sinal de fase e sinal de rotação



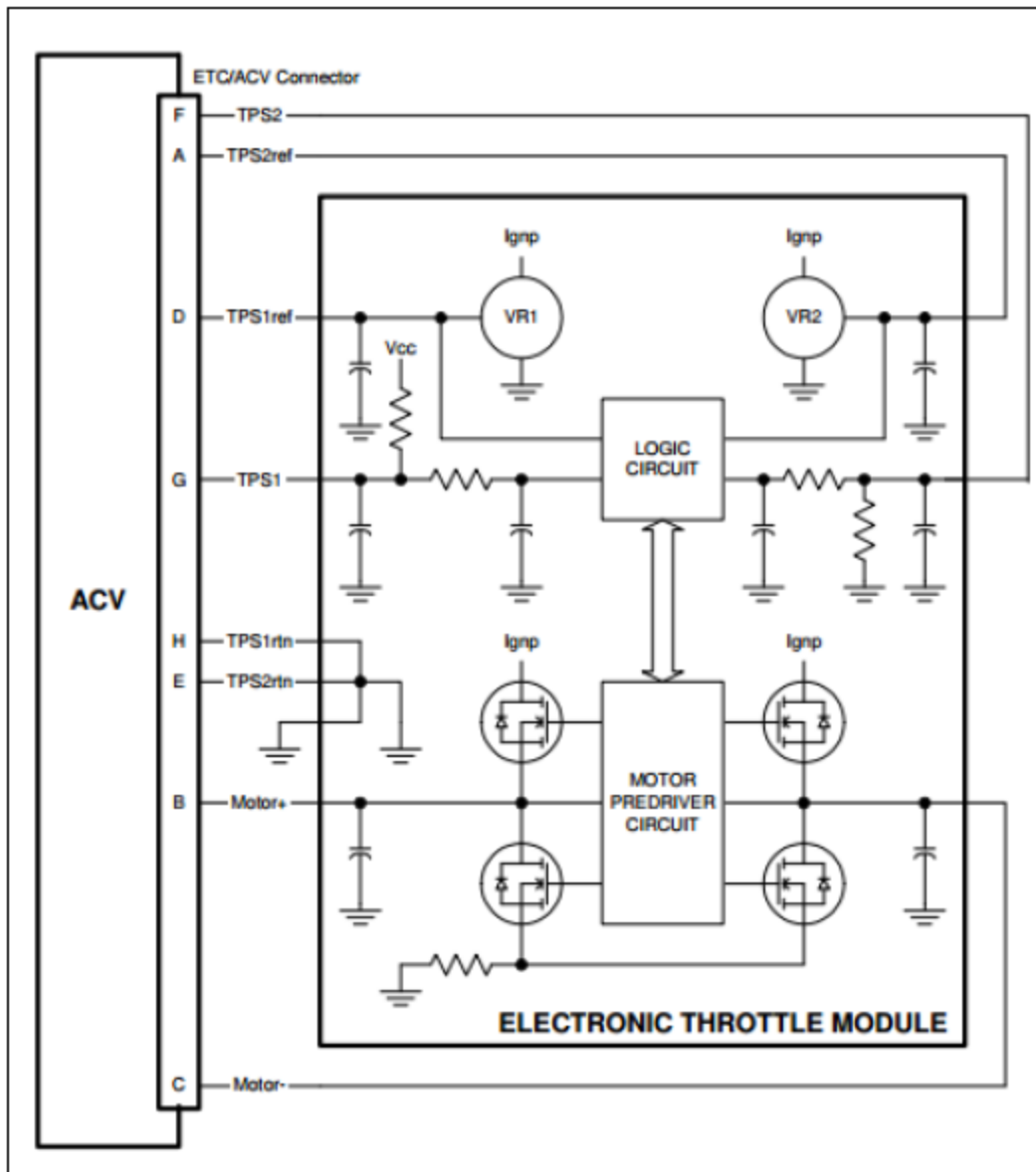
Fonte: Escola Politécnica da USP; Fatec Santo André (2013)

Válvula Borboleta

Figura 37 – Válvula Borboleta de Admissão de Ar



Fonte: o autor



Relés

Figura 39 – Relé de potência



Fonte: Bosch (2013c)

Válvulas Injetoras

Figura 40 – Válvula de injeção



Fonte: Bosch (2013a)

Bobinas de Ignição

Figura 41 – Tipos de bobinas de ignição



Fonte: Bosch (2013a)

3.1.3 O Motor Volkswagen 2.0L

Figura 42 – Motor Volkswagen 2.0L



Fonte: o autor

O motor a ser controlado pela unidade eletrônica deste projeto corresponde a um Volkswagen 2.0L 8v, equipado em um veículo modelo Polo Sedan 2004. Dentre as diversas características do motor, destacam-se:

- Operação com gasolina;
- Cilindrada de 1984 cm^3 (PORTAL VRUM, 2013);
- 116 cv de potência máxima a 5200 RPM (PORTAL VRUM, 2013);
- Torque máximo de 17,3 mkgf a 2400 RPM (PORTAL VRUM, 2013);
- 4 cilindros em linha (PORTAL VRUM, 2013);
- 2 válvulas por cilindro, característico do motor 8 válvulas (8v) (PORTAL VRUM, 2013);
- 2 bobinas de ignição dupla, com acionamento banco-a-banco (acionamento simultâneo nos cilindros 1-4 e 2-3, com "centelha perdida" em um dos cilindros);
- Injeção multiponto sequencial 1-3-4-2 (PORTAL VRUM, 2013).

3.1.4 A plataforma de projeto

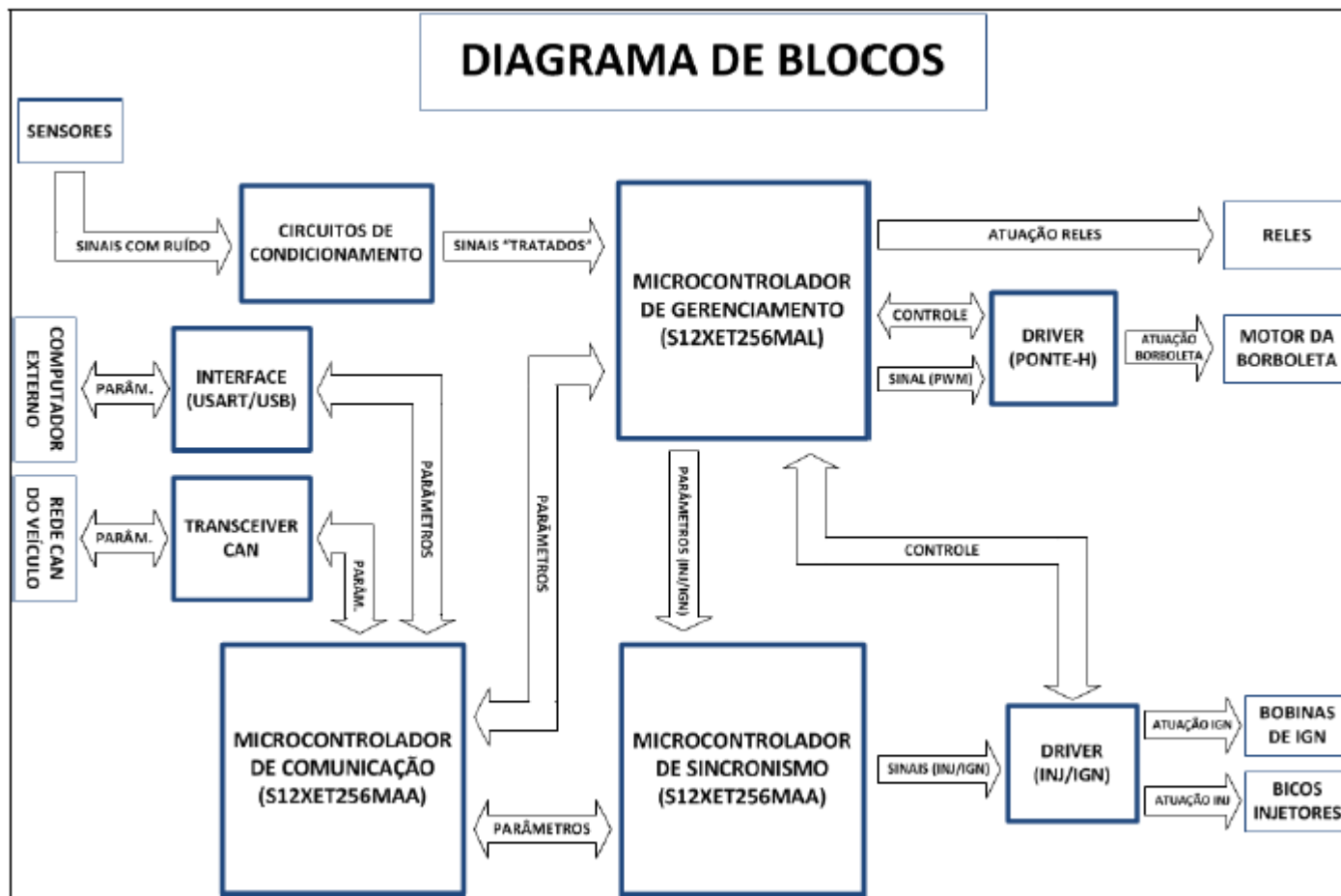
Figura 43 – Plataforma de trabalho para o desenvolvimento do *firmware*



Fonte: o autor

3.2.1 Hardware

Figura 44 – Diagrama de blocos do hardware

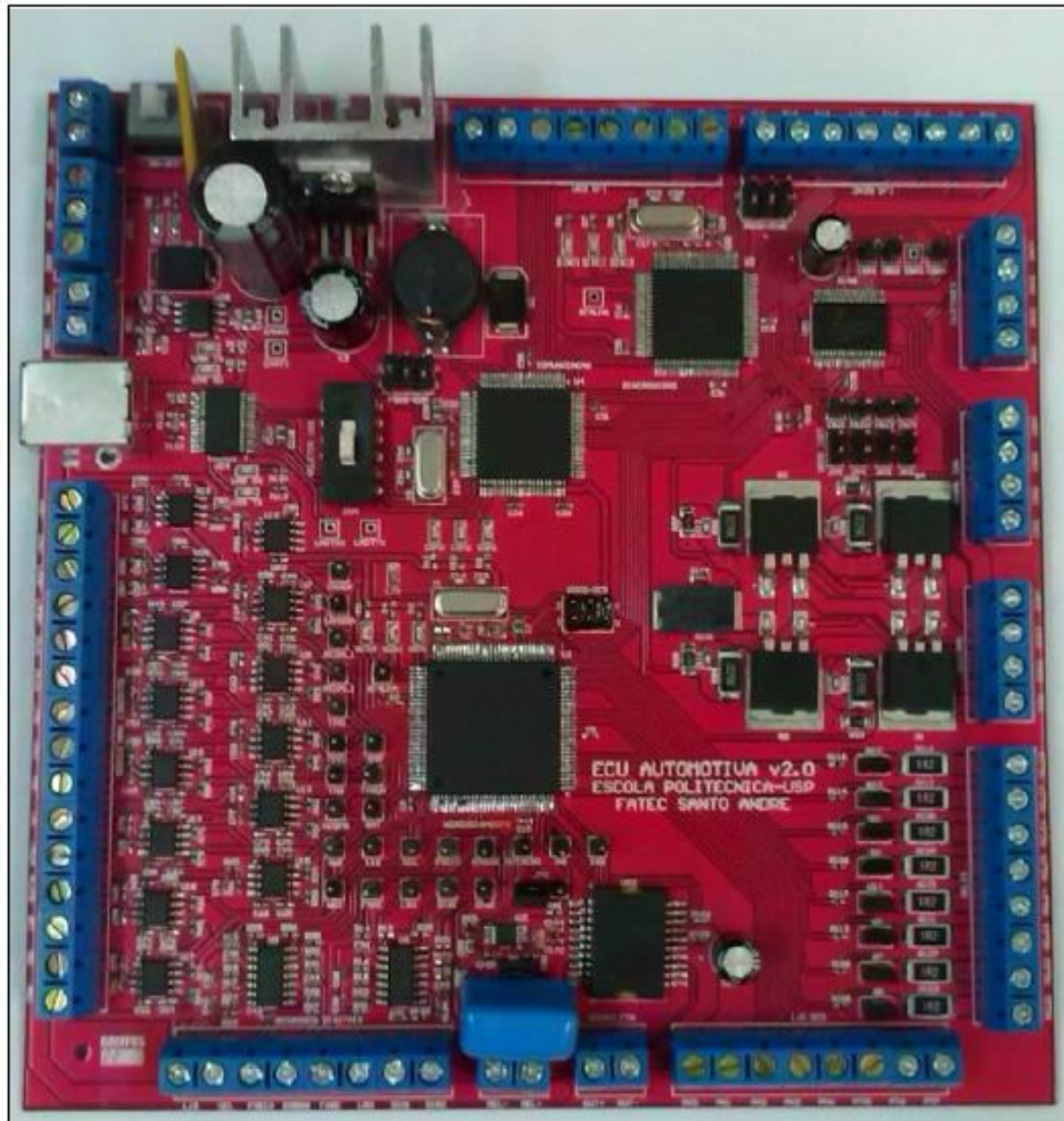


Fonte: o autor

- Gerenciamento: Leitura de sensores e cálculo dos parâmetros de atuação para injeção, ignição, válvula borboleta e relés;
- Sincronismo: Geração dos comandos de atuação para injeção e ignição (com parâmetros recebidos do Gerenciamento via protocolo SPI);
- Comunicação: Coleta periódica de dados dos blocos de Gerenciamento e Sincronismo, e envio dos mesmos para um computador externo (via protocolo USB) e para o painel do motorista do veículo (via protocolo CAN), além da leitura de velocidade do veículo (via protocolo CAN).

- Emprego de microcontroladores automotivos Freescale da família S12XE, que são mais robustos e mais eficientes que o PIC18F da Microchip, sendo que o S12XE é projetado para aplicações de gerenciamento de motores automotivos (FREESCALE, 2013);
- Uso de componentes SMD menores (incluindo resistores e capacitores), o que acarretou na redução da placa de 20cm x 20cm para 15cm x 15cm;
- Uso de filtros analógicos e comparadores de tensão para o tratamento de sensores analógicos e digitais;
- Remoção do multiplexador analógico 33972, dado que o microcontrolador utilizado neste projeto possui um número elevado de entradas analógicas;
- Emprego de circuitos de proteção contra sobretensão e curto-circuito;
- *Layout* de circuito impresso mais compacto, com uso de uma camada para aterramento;
- Adição de uma comunicação USB 2.0 com um computador externo;
- Retirada do *display* LCD¹⁴;
- Emprego de um *transceiver* CAN mais robusto;
- Adicionado pinos de medição dos sinais de sensores e atuadores;

Figura 45 - Protótipo da ECU v2.0



3.2.1.1 Microcontroladores

Figura 46 – S12XE com encapsulamento de 112 pinos

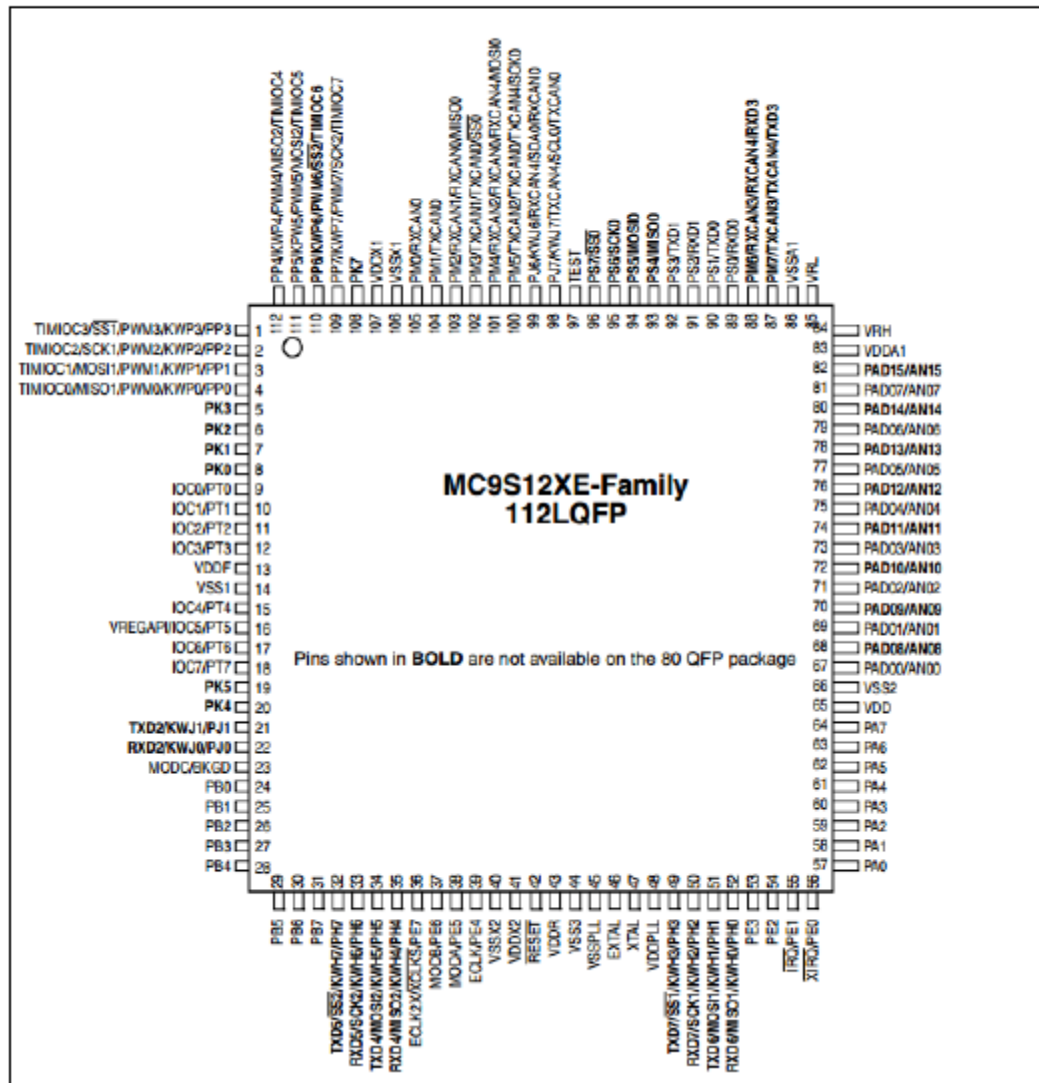
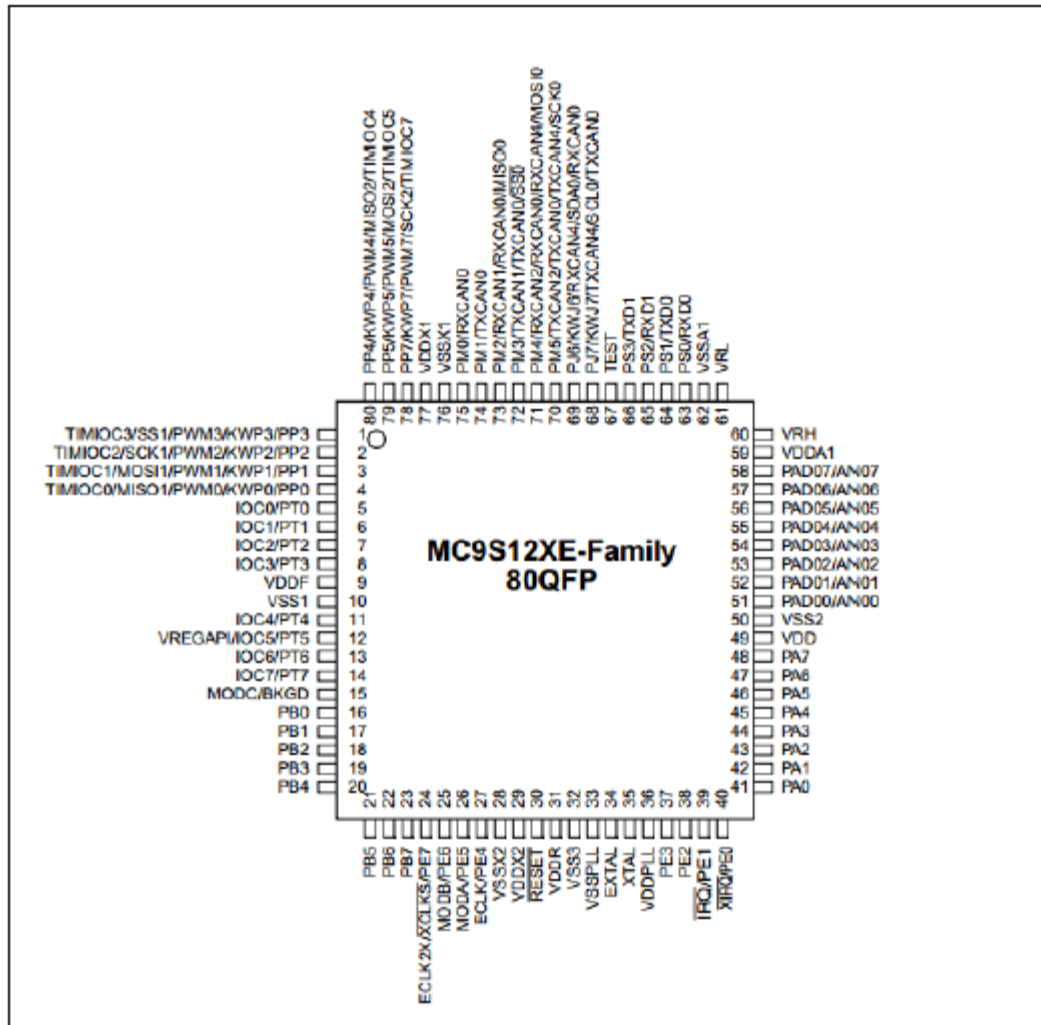


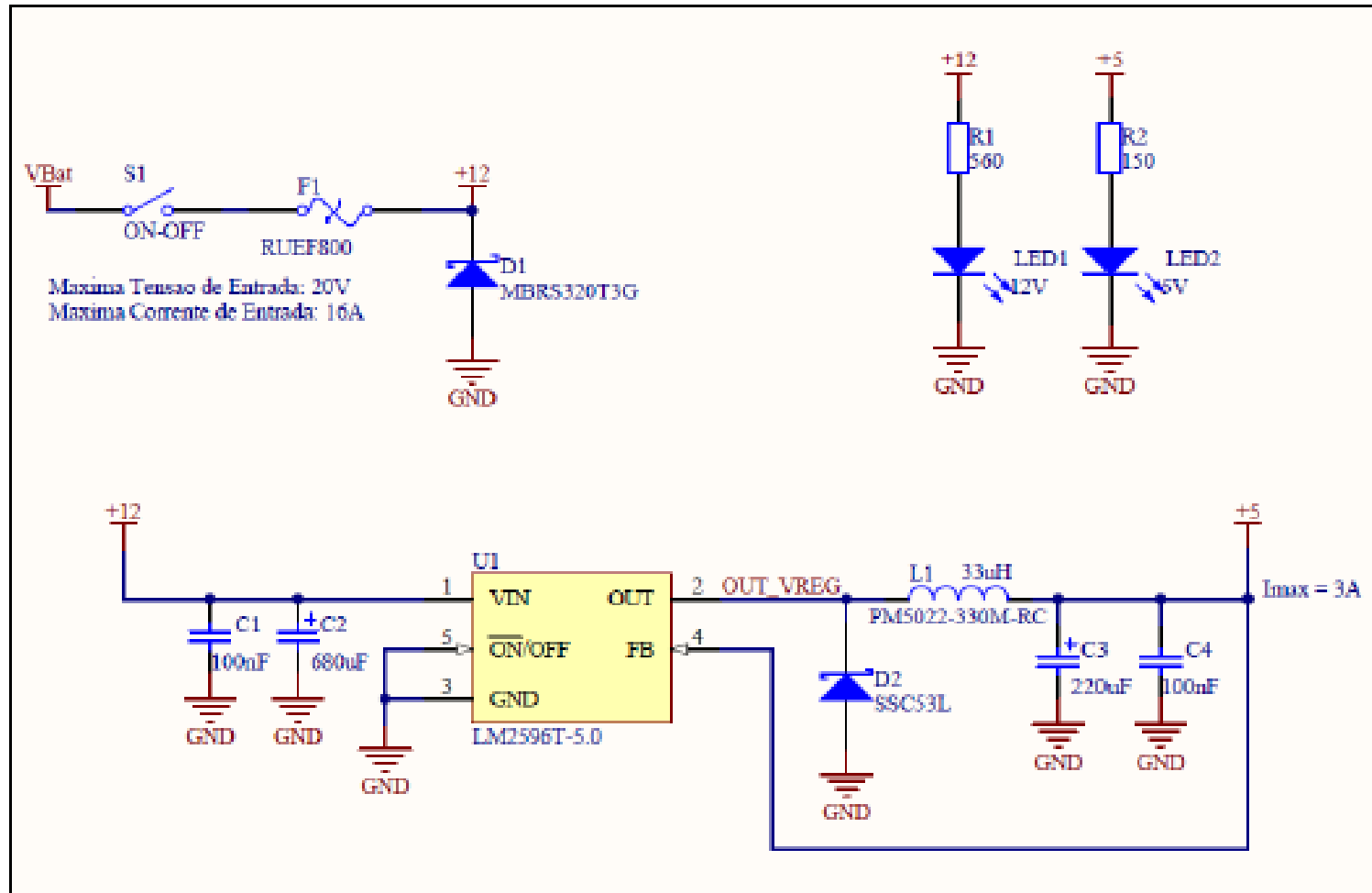
Figura 47 – S12XE com encapsulamento de 80 pinos



Fonte: Freescale (2012)

3.2.1.2 Regulador de Tensão

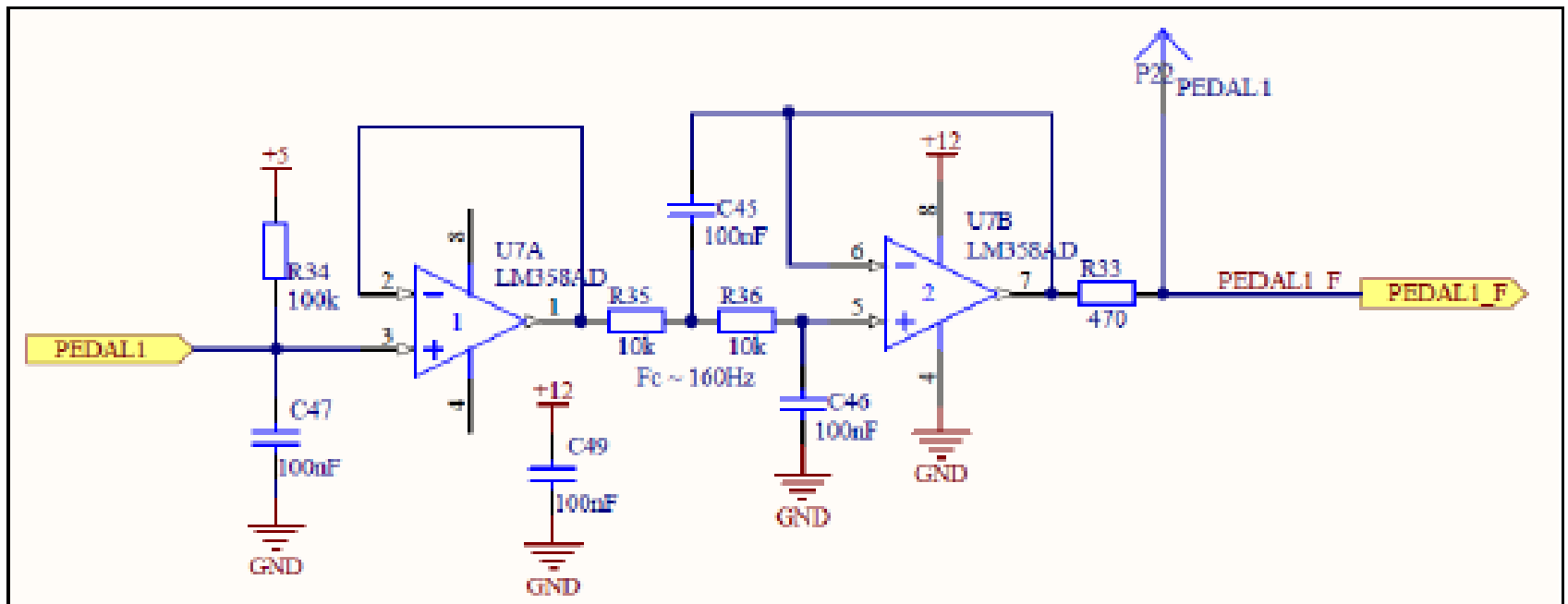
Figura 48 – Esquema elétrico do regulador de tensão



Fonte: o autor

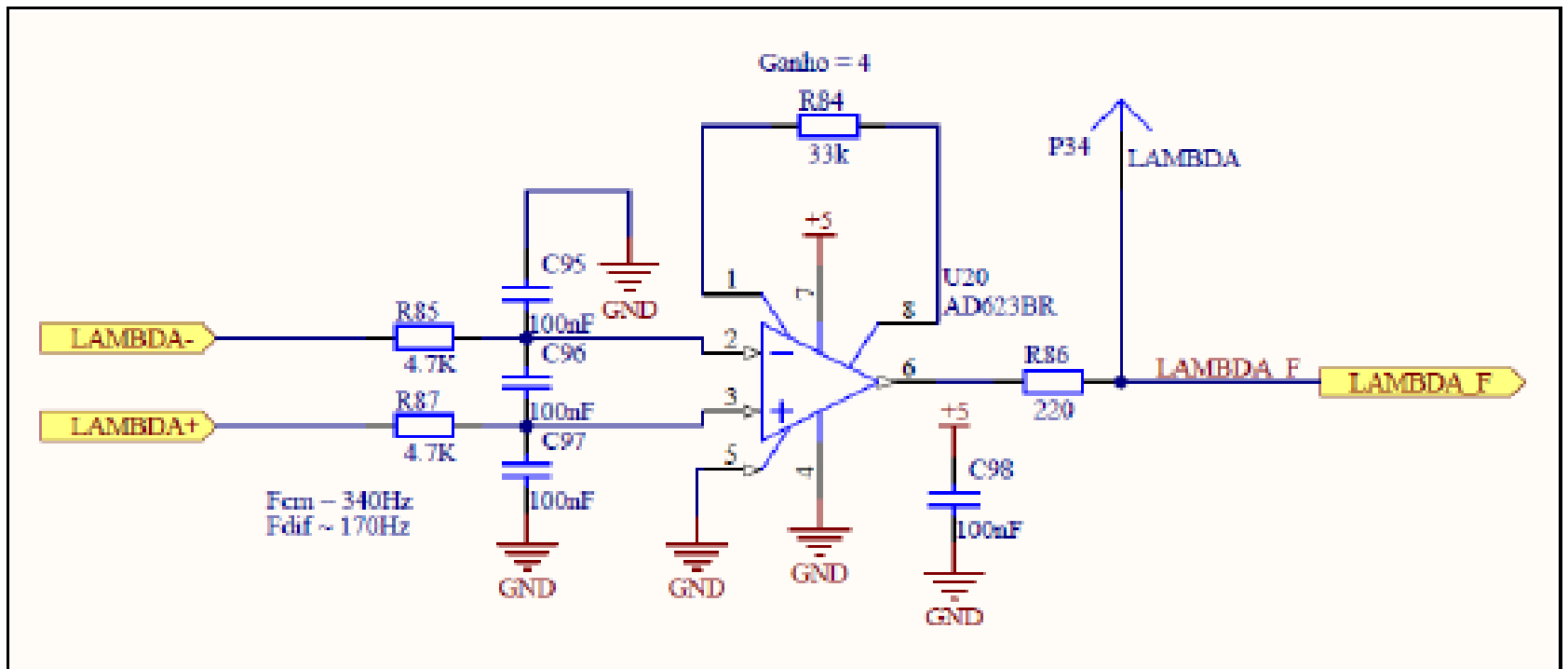
3.2.1.3 Condicionamento dos Sensores Analógicos

Figura 49 – Esquema elétrico do filtro analógico de segunda ordem



Fonte: o autor

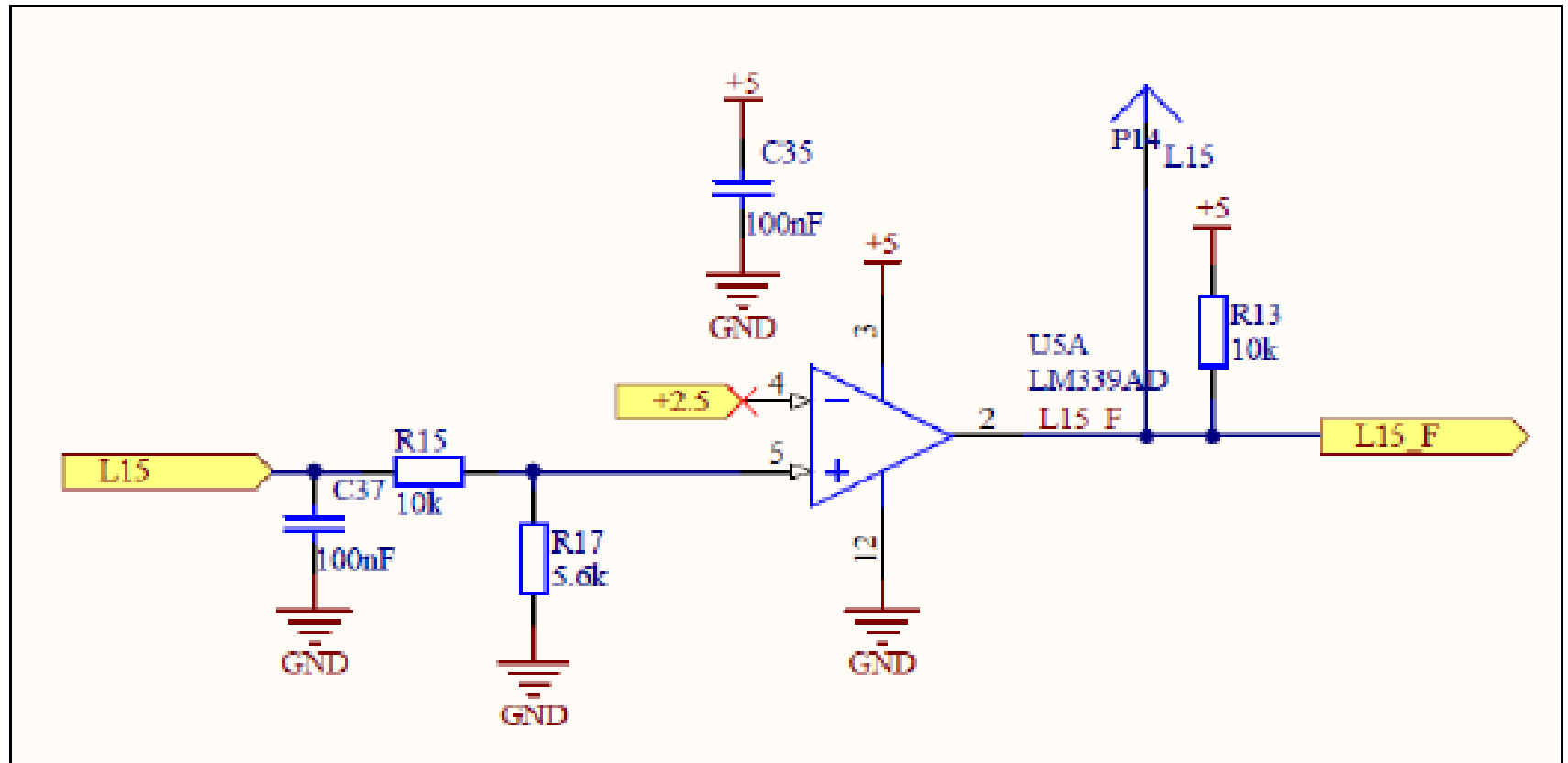
Figura 50 – Esquema elétrico do amplificador de instrumentação



Fonte: o autor

3.2.1.4 Condicionamento dos Sensores Digitais

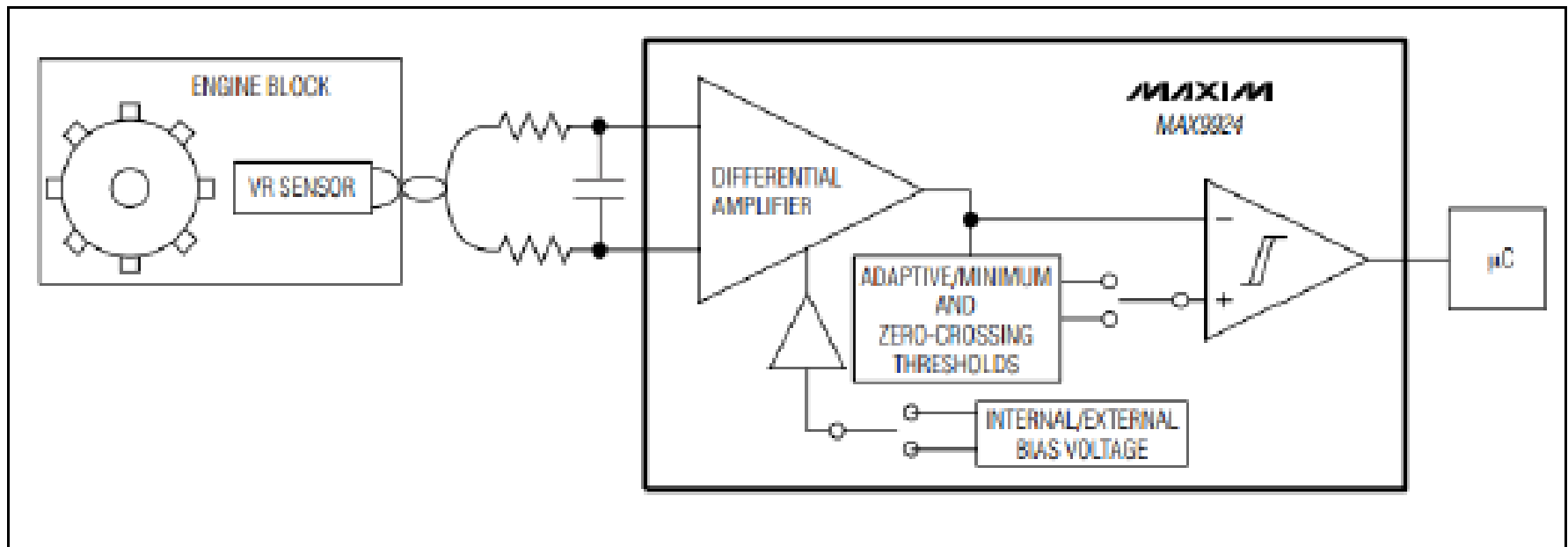
Figura 51 – Esquema elétrico do comparador de tensão



Fonte: o autor

3.2.1.5 Condicionamento do Sinal de Rotação

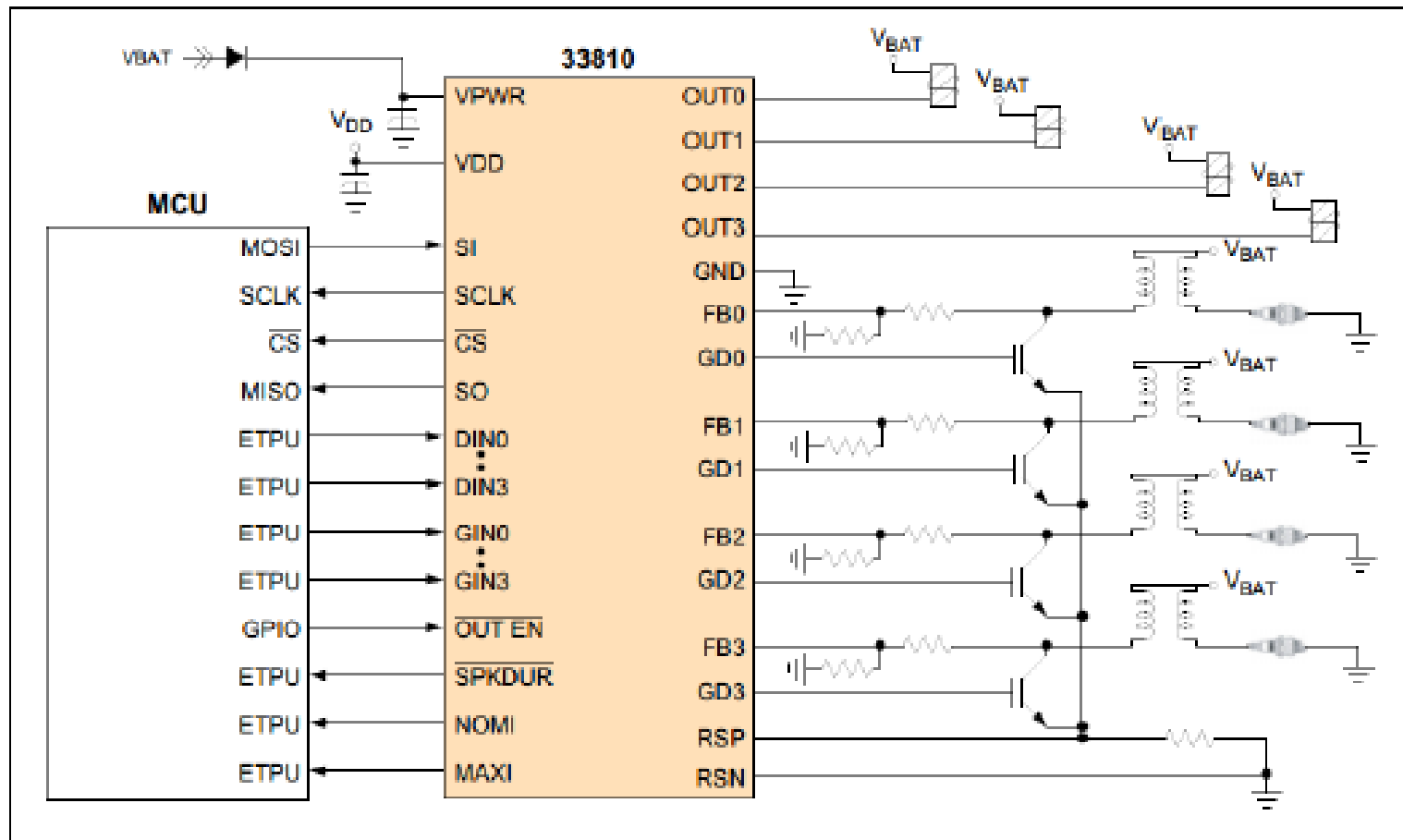
Figura 52 – Exemplo de aplicação do circuito integrado MAX9924



Fonte: Maxim Integrated Products (2012)

3.2.1.6 Driver de Injeção e Ignição

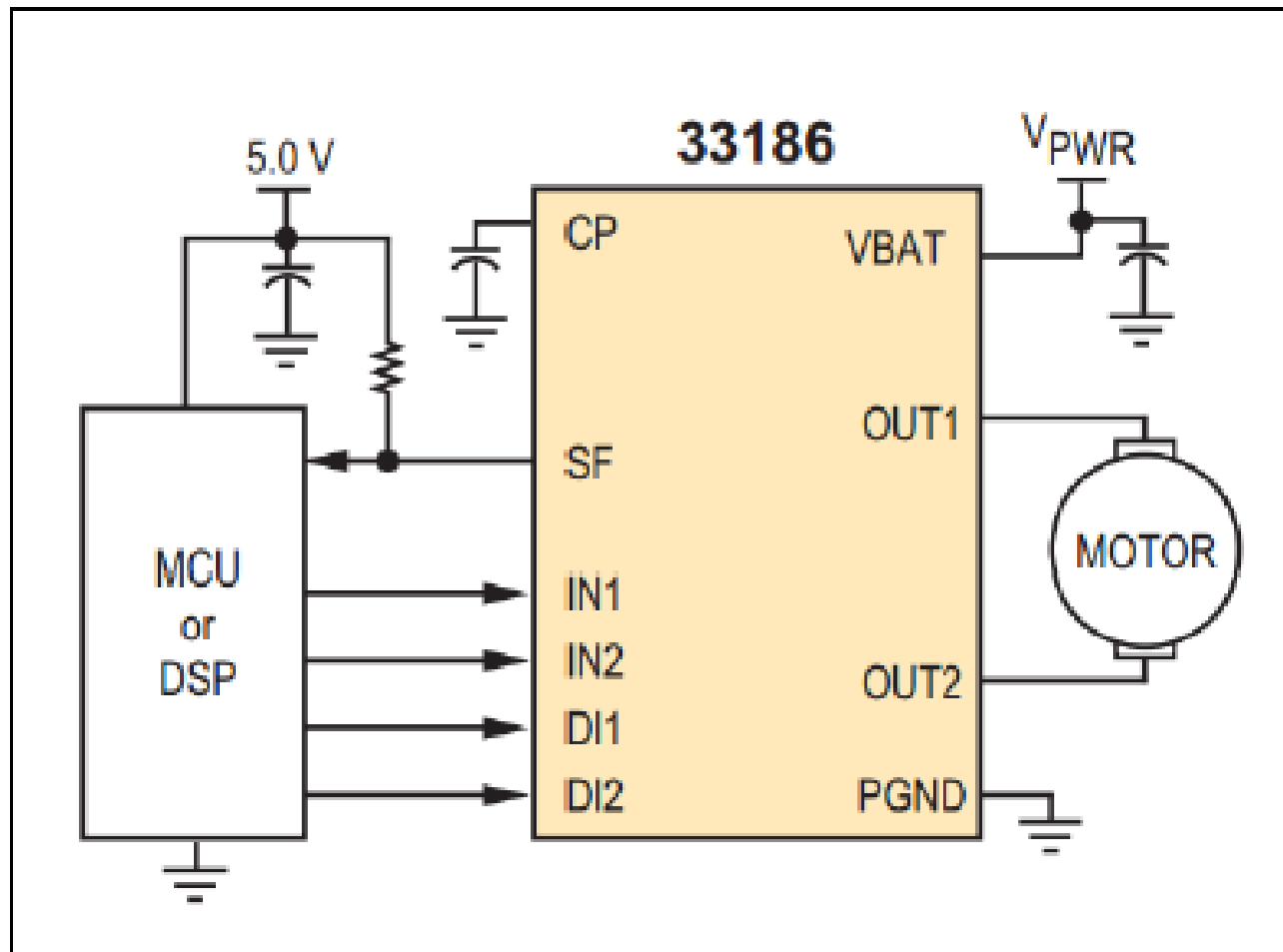
Figura 53 – Exemplo de aplicação do CI 33810



Fonte: Freescale (2008)

Driver para a Válvula Borboleta (Ponte H)

Figura 54 – Exemplo de aplicação do CI 33186



Fonte: Freescale (2011)

Figura 55 – Tabela da verdade do CI 33186

Device State	Input Conditions				Status		Outputs	
	DI1 ⁽⁸⁾	DI2 ⁽⁸⁾	IN1	IN2	SF ⁽⁹⁾	SF ⁽¹⁰⁾	OU1	OU2
1-Forward	L	H	H	L	H	H	H	L
2-Reverse	L	H	L	H	H	H	L	H
3-Free Wheeling Low	L	H	L	L	H	H	L	L
4-Free Wheeling High	L	H	H	H	H	H	H	H
5-Disable 1	H	X	X	X	L	H	Z	Z
6-Disable 2	X	L	X	X	L	H	Z	Z
7-IN1 Disconnected	L	H	Z	X	H	H	H	X
8-IN2 Disconnected	L	H	X	Z	H	H	X	H
9-DI1 Disconnected	Z	X	X	X	L	H	Z	Z
10-DI2 Disconnected	X	Z	X	X	L	H	Z	Z
11-Current Limit.active	L	H	X	X	H	H	Z	Z
12-Undervoltage ⁽⁶⁾	X	X	X	X	L	L	Z	Z
13-Overtemperature ⁽⁷⁾	X	X	X	X	L	L	Z	Z
14-Overcurrent ⁽⁷⁾	X	X	X	X	L	L	Z	Z

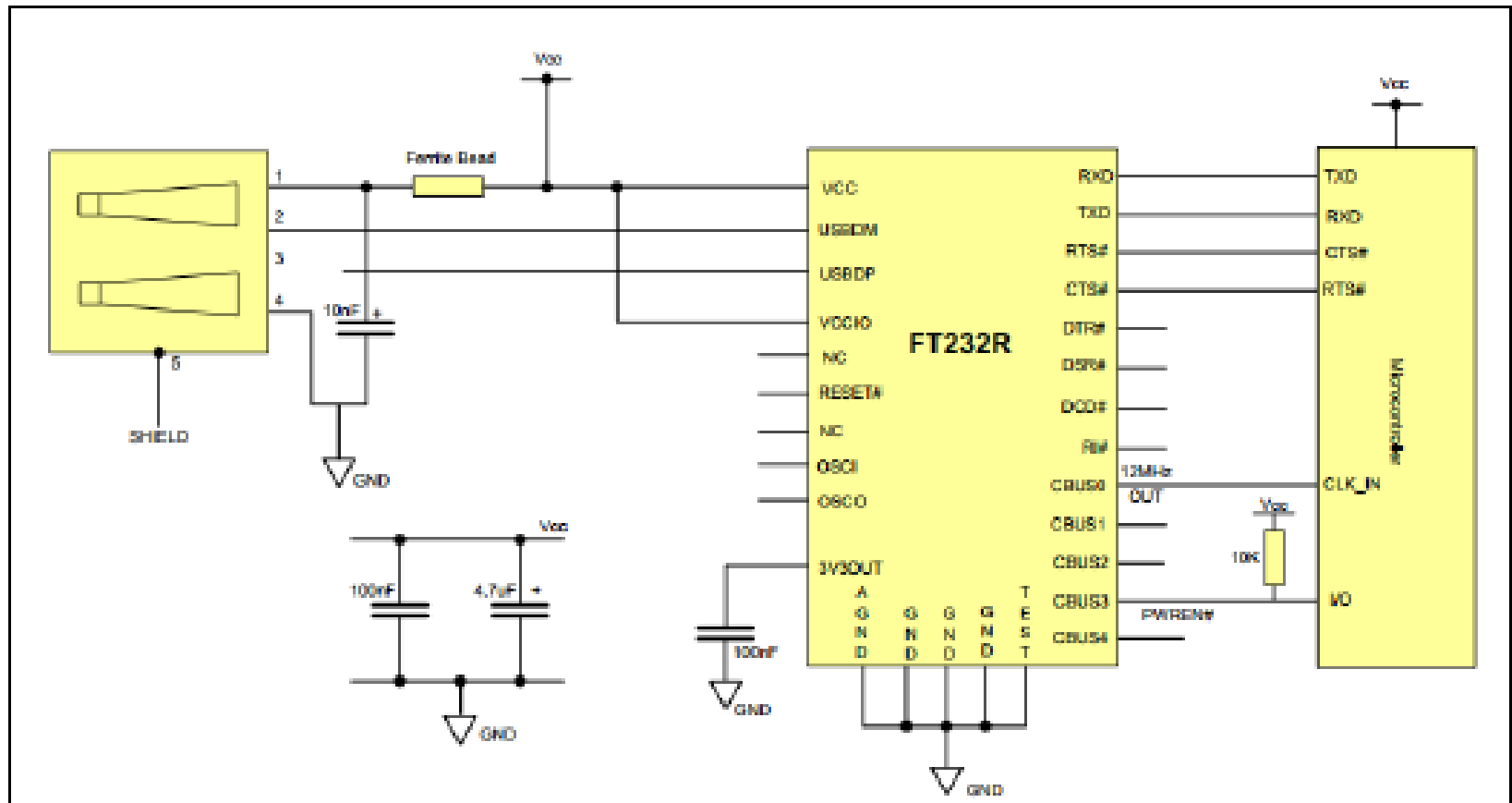
Notes

- In case of undervoltage, tristate and status-flag are reset automatically.
- Whenever overcurrent or overtemperature is detected, the fault is stored (i.e. status-flag remains low). The tristate conditions and the status-flag are reset via DI1 (IN1) or DI2 (IN2). Pin names in brackets refer to coding Pin (COD = VCC).
- If COD = VCC then DI1 and DI2 are not active.
- COD = nc or GND
- COD = VCC

L = Low
H = High
X = High or Low
Z = High impedance (all output stage transistors are switched off).

3.2.1.8 Conversor UART-USB

Figura 56 – Exemplo de aplicação do FT232 com alimentação via USB



Fonte: Future Technology Devices International Limited (2010)

3.2.1.11 Pinagem da placa

A placa foi projetada para facilitar as conexões elétricas com os sinais provenientes do motor. Para isto, escritas de identificação foram colocadas próximas de cada pino. Abaixo segue uma descrição de cada conector da placa.

Tabela 2 – Pinagem do *hardware* da ECU v2.0

Conector	Pino	E/S	Descrição	Observação
VIN	VBAT	E	Positivo da Bateria	
	GND	E	Negativo da Bateria (massa)	
VOUT	12V	S	Tensão da Bateria (aprox. 12V)	
	5V	S	Tensão de 5V controlada	
	GND	E	Massa	
CAN	CANL	E/S	Comunicação CAN	
	CANH	E/S	Comunicação CAN	
USB	-	E/S	Comunicação USB	
SENSORES ANALÓGICOS	KNO-	E	Negativo do sensor Knock	Não utilizado
	KNO+	E	Positivo do sensor Knock	Não utilizado
	LAMB-	E	Negativo da sonda Lambda	Não utilizado
	LAMB+	E	Positivo da sonda Lambda	Não utilizado
	PED1	E	Pedal 1	
	PED2	E	Pedal 2	Não utilizado
	TPS1	E	TPS1 da borboleta	Não utilizado
	TPS2	E	TPS2 da borboleta	
	MAP	E	MAP	
	TAR	E	Temperatura do ar	
	TAGUA	E	Temperatura da água	
	NCOMB	E	Nível de combustível	Não utilizado
	BAT	E	Nível da Bateria	
	AN1	E	Canal analógico reserva 1	Não utilizado
AN2	E	Canal analógico reserva 2	Não utilizado	

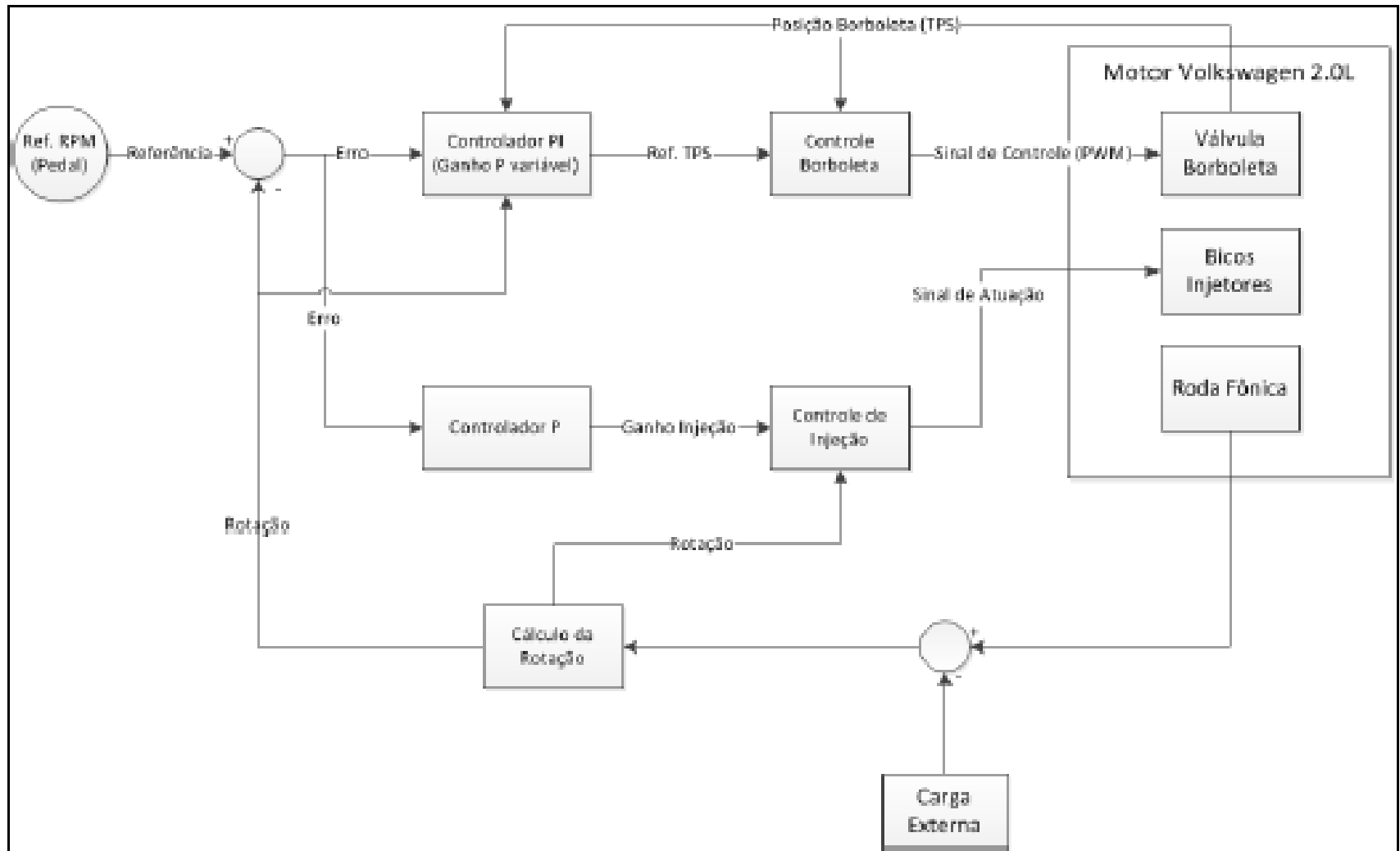
	L15	E	Linha 15	
	VEL	E	Velocidade	Não utilizado
	FREIO	E	Freio	Não utilizado
SENSORES	EMBR	E	Embreagem	Não utilizado
DIGITAIS	FASE	E	Fase	
	L50	E	Linha 50	Não utilizado
continuação da página anterior				
	DIG1	E	Canal digital reserva 1	Não utilizado
	DIG2	E	Canal digital reserva 2	Não utilizado
SENSOR	REL-	E	Negativo do sensor de relutância v.	
	REL+	E	Positivo do sensor de relutância v.	
BORBOLETA	OUT+	S	Positivo do motor da borboleta	
	OUT-	S	Negativo do motor da borboleta	
RELES	RELE1	S	Acionamento do Relé 1	
	RELE2	S	Acionamento do Relé 2	Não utilizado
	RELE3	S	Acionamento do Relé 3	Não utilizado
	RELE4	S	Acionamento do Relé 4	Não utilizado
	RELE5	S	Acionamento do Relé 5	Não utilizado
	RELE6	S	Acionamento do Relé 6	Não utilizado
	RELE7	S	Acionamento do Relé 7	Não utilizado
	RELE8	S	Acionamento do Relé 8	Não utilizado
BOBINAS	BOB1	S	Acionamento da bobina de ign 1	Não utilizado
	BOB2	S	Acionamento da bobina de ign 2	Não utilizado
	BOB3	S	Acionamento da bobina de ign 3	Não utilizado
	BOB4	S	Acionamento da bobina de ign 4	Não utilizado

SINAL IGN	IGN1	S	Acionamento do driver de ign 1	
	IGN2	S	Acionamento do driver de ign 2	
	IGN3	S	Acionamento do driver de ign 3	
	IGN4	S	Acionamento do driver de ign 4	
INJETORES	INJ1	S	Acionamento do bico injetor 1	
	INJ2	S	Acionamento do bico injetor 2	
	INJ3	S	Acionamento do bico injetor 3	
	INJ4	S	Acionamento do bico injetor 4	
I/O GER	PK0	E/S	Pino 0 da porta K do Ger.	Portas extras disponibilizadas para eventuais conexões
	PK1	E/S	Pino 1 da porta K do Ger.	
	PK2	E/S	Pino 2 da porta K do Ger.	
	PK3	E/S	Pino 3 da porta K do Ger.	
	PP6	E/S	Pino 6 da porta P do Ger.	
	PT5	E/S	Pino 5 da porta T do Ger.	
	PT6	E/S	Pino 6 da porta T do Ger.	
I/O SINC	PT7	E/S	Pino 7 da porta T do Ger.	Portas extras disponibilizadas para eventuais conexões
	PT0	E/S	Pino 0 da porta B* do Sinc.	
	PT1	E/S	Pino 1 da porta B* do Sinc.	
	PT2	E/S	Pino 2 da porta B* do Sinc.	
	PT3	E/S	Pino 3 da porta B* do Sinc.	
	PT4	E/S	Pino 4 da porta B* do Sinc.	
	PT5	E/S	Pino 5 da porta B* do Sinc.	

continua na próxima página

continuação da página anterior				
	PT6	E/S	Pino 6 da porta B* do Sinc.	*errata da placa
	PT7	E/S	Pino 7 da porta B* do Sinc.	
I/O COM	PT0	E/S	Pino 0 da porta T do Com.	
	PT1	E/S	Pino 1 da porta T do Com.	
	PT2	E/S	Pino 2 da porta T do Com.	Portas extras
	PT3	E/S	Pino 3 da porta T do Com.	disponibilizadas
	PT4	E/S	Pino 4 da porta T do Com.	para eventuais
	PT5	E/S	Pino 5 da porta T do Com.	conexões
	PT6	E/S	Pino 6 da porta T do Com.	
	PT7	E/S	Pino 7 da porta T do Com.	
PROG-GER	-	E	Programação in-circuito do GER.	
PROG-SINC	-	E	Programação in-circuito do SINC.	
PROG-COM	-	E	Programação in-circuito do COM.	

Figura 59 – Diagrama do controle de Rotação



Fonte: o autor

Abaixo foi destacado um trecho do código do uC de Gerenciamento que implementa o controle de rotação:

```
void controle_rpm() {
    float control = 0;
    float ref_vb_temp, ref_vb_base;

    // Algoritmo de controle PI
    erro_rpm = ref_rpm - (float)rotacao;

    if ((erro_rpm < 70.0) && (erro_rpm > -70.0))
        return;

    // Atualiza a somatoria
    if ((erro_rpm < 500.0) && (erro_rpm > -500.0)) // Integracao
        condicional - Anti wind-up
        soma_erro_rpm += erro_rpm;
    else
        soma_erro_rpm = 0; // Descarrega a somatoria

    // Anti wind-up (Limita a somatoria)
    if (soma_erro_rpm > 200000.0)
        soma_erro_rpm = 200000.0;
    else if (soma_erro_rpm < -200000.0)
        soma_erro_rpm = -200000.0;

    if (erro_rpm > 0.0) { // Aceleracao
        if (modo_operacao == 0) { // Modo de operacao "normal"
            if (rotacao < 1700)
                Kp_rpm = erro_rpm*0.08/5000.0; // Ajustar o primeiro
                parametro da formula
            else if (rotacao < 3000)
                Kp_rpm = erro_rpm*(0.00013*(float)rotacao-0.142)/5000.0;
                // Ajustar o primeiro parametro da formula
            else
```

atua como distúrbio no controle, de maneira a reduzir a rotação. Assim, cabe ao controle de rotação atuar de maneira a impedir esta redução, o que, conseqüentemente, fornece torque ao eixo do motor. Já o controle de posição da borboleta é realizado através de um controlador PI (proporcional e integral).

Abaixo foi destacado um trecho do código do uC de Gerenciamento que implementa o controle de rotação:

```
void controle_rpm() {
    float control = 0;
    float ref_vb_temp, ref_vb_base;

    // Algoritmo de controle PI
    erro_rpm = ref_rpm - (float)rotacao;

    if ((erro_rpm < 70.0) && (erro_rpm > -70.0))
        return;

    // Atualiza a somatoria
    if ((erro_rpm < 500.0) && (erro_rpm > -500.0)) // Integracao
        condicional - Anti wind-up
        soma_erro_rpm += erro_rpm;
    else
        soma_erro_rpm = 0; // Descarrega a somatoria

    // Anti wind-up (Limita a somatoria)
    if (soma_erro_rpm > 200000.0)
        soma_erro_rpm = 200000.0;
    else if (soma_erro_rpm < -200000.0)
        soma_erro_rpm = -200000.0;

    if (erro_rpm > 0.0) { // Aceleracao
        if (modo_operacao == 0) { // Modo de operacao "normal"
            if (rotacao < 1700)
                Kp_rpm = erro_rpm*0.08/5000.0; // Ajustar o primeiro
                parametro da formula
            else if (rotacao < 3000)
                Kp_rpm = erro_rpm*(0.00013*(float)rotacao-0.142)/5000.0;
                // Ajustar o primeiro parametro da formula
            else
                Kp_rpm = erro_rpm*0.25/5000.0; // Ajustar o primeiro
                parametro da formula
        }
        else { // Modo de operacao "economico" (ganhos menores)
            /*if (rotacao < 1700)
                Kp_rpm = erro_rpm*0.04/5000.0; // Ajustar o primeiro
                parametro da formula
            else if (rotacao < 3000)
```

```

        Kp_rpm = erro_rpm*(0.00007*(float)rotacao-0.078)/5000.0;
        // Ajustar o primeiro parametro da formula
    else
        Kp_rpm = erro_rpm*0.13/5000.0; // Ajustar o primeiro
        parametro da formula*/
        Kp_rpm = erro_rpm*0.01/5000.0;
    }
}

else // Desaceleracao - Nao usa ganho Kp
    Kp_rpm = 0.0;

control = Kp_rpm*erro_rpm + Ki_rpm*soma_erro_rpm;
ref_vb_base = 0.0054*ref_rpm + 32.07; // Posicao da vb para manter a
    rotacao
ref_vb_temp = ref_vb_base + control;

// Limitacao da posicao max da vb para o motor nao "afogar" em
    aceleracoes grandes
//if (rotacao < 1000.0)
//    ref_vb_max = 67.3;
if (rotacao < 1700)
    ref_vb_max = 137.0;
else {
    ref_vb_max = (TPS_MAX - 137.0)*((float)rotacao - 1700.0)/1300.0
        + 137.0;
    if (ref_vb_max > TPS_MAX)
        ref_vb_max = TPS_MAX;
}

// Limita referencia para controle da borboleta
if ((ref_vb_temp < TPS_1000) && (rotacao > 1500))
    ref_vb_temp = TPS_1000;
else if (ref_vb_temp < 32.0) // Impede fechamento total da vb
    ref_vb_temp = 32.0;
else if (ref_vb_temp > ref_vb_max) // Limitacao para o motor nao "
    afogar" em aceleracoes grandes
    ref_vb_temp = ref_vb_max;

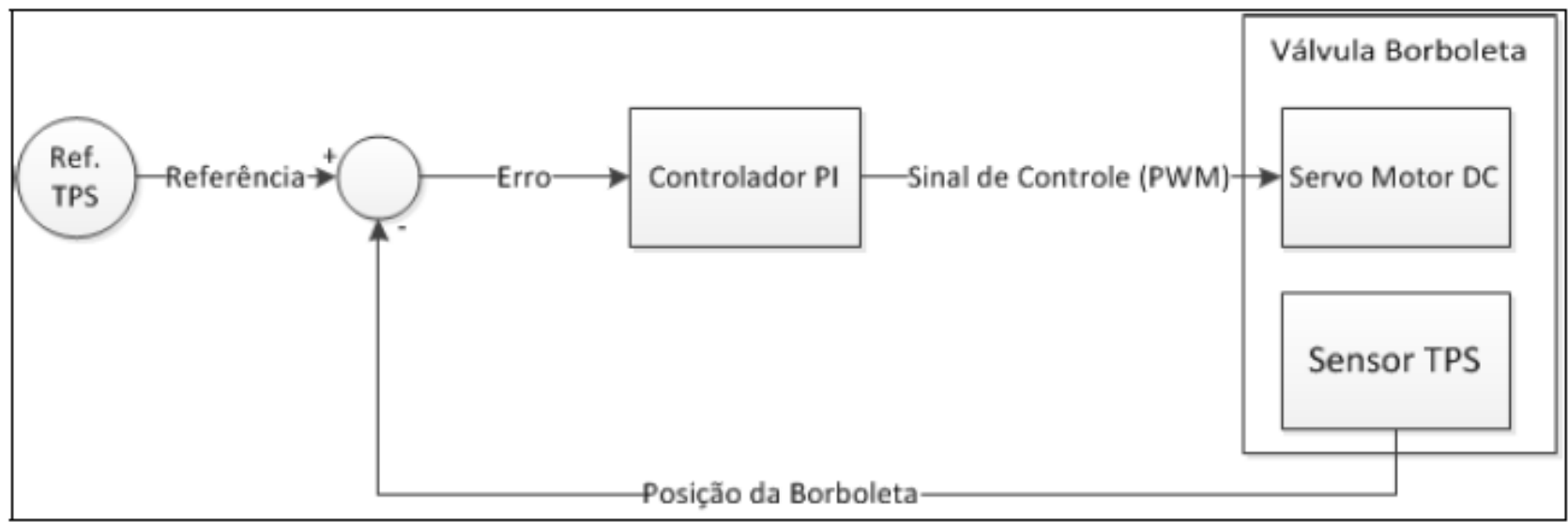
if (((erro_rpm > 1000.0) || (erro_rpm < -1000.0)) && (rotacao <
    1000))
    while ((injecao_ok < 1) && (reset == 0)); // Espera injetar
        antes de abrir a vb...

ref_vb = ref_vb_temp; // Passa p/ a variavel final

injecao_ok = 0; // reseta a variavel

```

Figura 60 – Diagrama do controle de posição da Válvula Borboleta



Fonte: o autor

```

interrupt void TIMERO_CONTROLE_VB_XGATE(void)
{
    float control; // Sinal de controle
    unsigned char pwm; // Valor de PWM a ser gerado
    unsigned char inv = 0;

    PITTF = (1<<0); // Limpa flag PTFO

    if (ref_vb < 10.0)
        return; // Nao Atua na borboleta

    IOGER7 = 1; // PARA TESTES

    /*-----Controle da Valvula Borboleta
    -----*/

    // Algoritmo de controle PI
    erro_vb = ref_vb - (float)tps2;

    //if ((erro_vb < 1.0) &&& (erro_vb > -1.0) &&& (tps2 < TPS_MIN)) {
    //    IOGER7 = 0;
    //    return;
    //}

    soma_erro_vb += erro_vb;

    control = Kp_vb*erro_vb + Ki_vb*soma_erro_vb;

    if (control > 0) { // Valvula precisa ser aberta
        inv = 0;
        if (control > 230.0) // Limitacao do sinal (PWM = 90%)
            control = 230.0;
    }

    else { // Valvula precisa ser fechada
        control = -control;
        inv = 1;
    }
}

```



```
pwm = (unsigned char)control; // Passa o calculo do pwm p/ a "
    variavel final"
```

```
if (!inv) {
    PWMDTY5 = pwm;
    PWMDTY4 = 0;
```

```
}
```

```
else {
    PWMDTY5 = 0;
    PWMDTY4 = pwm;
```

```
}
```

```
I0GER7 = 0;
```

```
}
```

```

// Calculo da rotacao
periodo_temp = (float)periodo_3 / 3.0; // Timer1 mediu a duracao de 3
      dentes
rotacao_temp = 1000000.0/periodo_temp;

rotacao = (unsigned int)rotacao_temp;

...

// Calculo da massa de ar admitida por cilindro
// OBS: Utiliza a Lei dos gases ideais (PV/4=nRT)
m_ar = 0.25 * pressao_ar * 1.984 * 28.96 / (8.314 * t_ar); // Unidade: g

// Calculo da massa de combustivel para lambda=1
m_comb = m_ar / 13.3; // Unidade: g

// Calculo do tempo de injecao (Fluzo: 176.93 g/min)
tempo_inj_temp = m_comb * 339117.17; // Unidade: us

// Ajuste do fator de injecao em caso de aceleracao/desaceleracao
fator_inj = 1.0 + Kp_inj*erro_rpm;

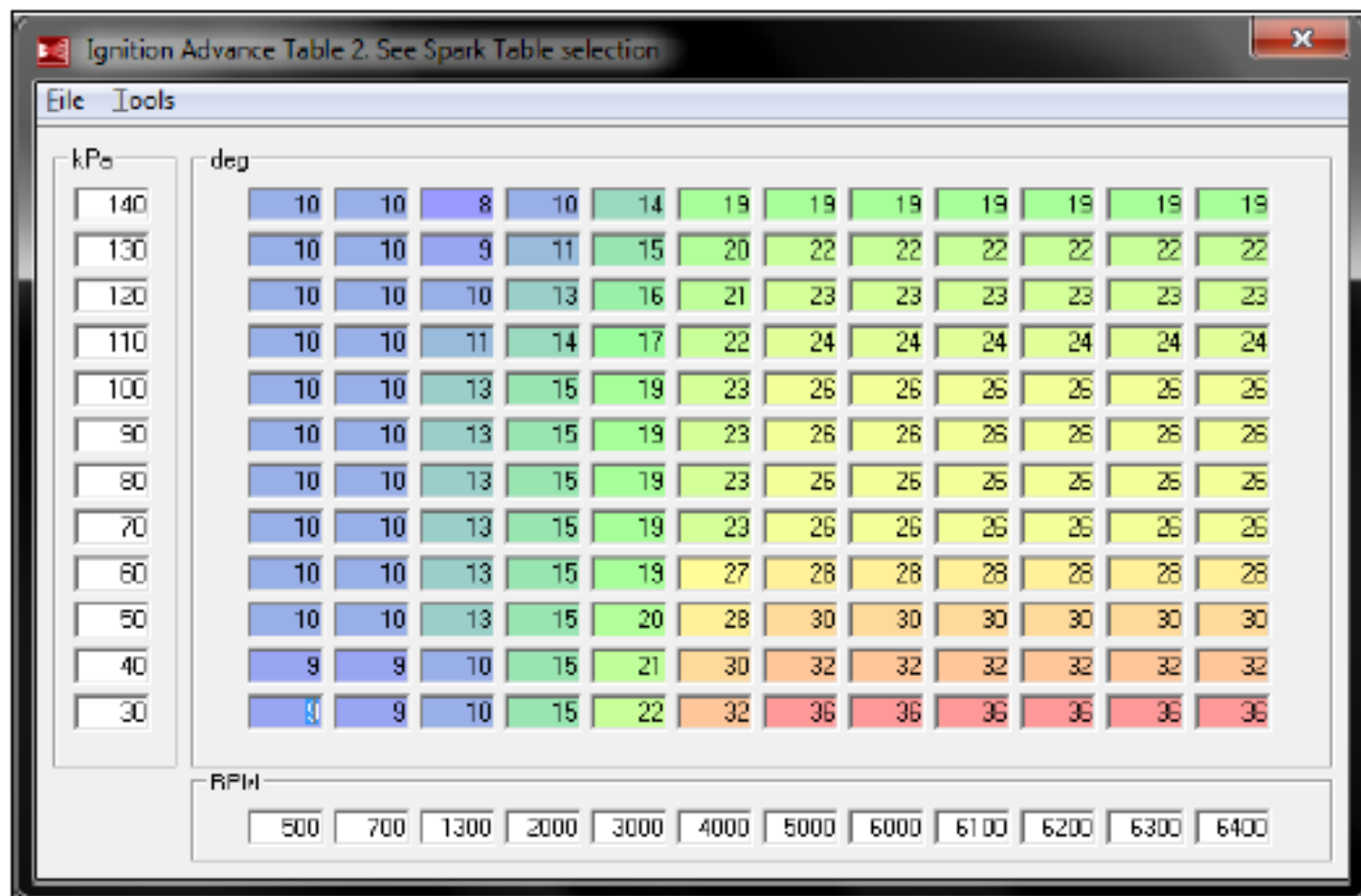
// Fator de correcao na injecao para motor frio
if (t_agua < 40.0)
      fator_inj += 0.2;
else if (t_agua < 70.0)
      fator_inj += 0.467 - 0.0067*t_agua;

// Limitacao superior do fator de injecao (equivale a lambda - 0.71)
if (fator_inj > 1.4)
      fator_inj = 1.4;

// Limitacao inferior do fator de injecao (equivale a lambda - 1.33)
else if (fator_inj < 0.75)
      fator inj = 0.75;

```

Figura 61 – Mapa de ignição de um motor Volkswagen 2.0L 8v



Fonte: ClubPolo forums (2013)

Tabela 4 – Mapa de ignição (em graus) com problema de detonação

RPM kPa	30	40	50	60	70	80	90	100	110	120	
500	9	9	10	10	10	10	10	10	10	10	
700	9	9	10	10	10	10	10	10	10	10	
1300	10	10	13	13	13	13	13	13	11	10	
continuação da página anterior											
2000	15	15	15	15	15	15	15	15	15	14	13
3000	22	21	20	19	19	19	19	19	19	17	16
4000	32	30	28	27	23	23	23	23	23	22	21
5000	36	32	30	28	26	26	26	26	26	24	23
6000	36	32	30	28	26	26	26	26	26	24	23

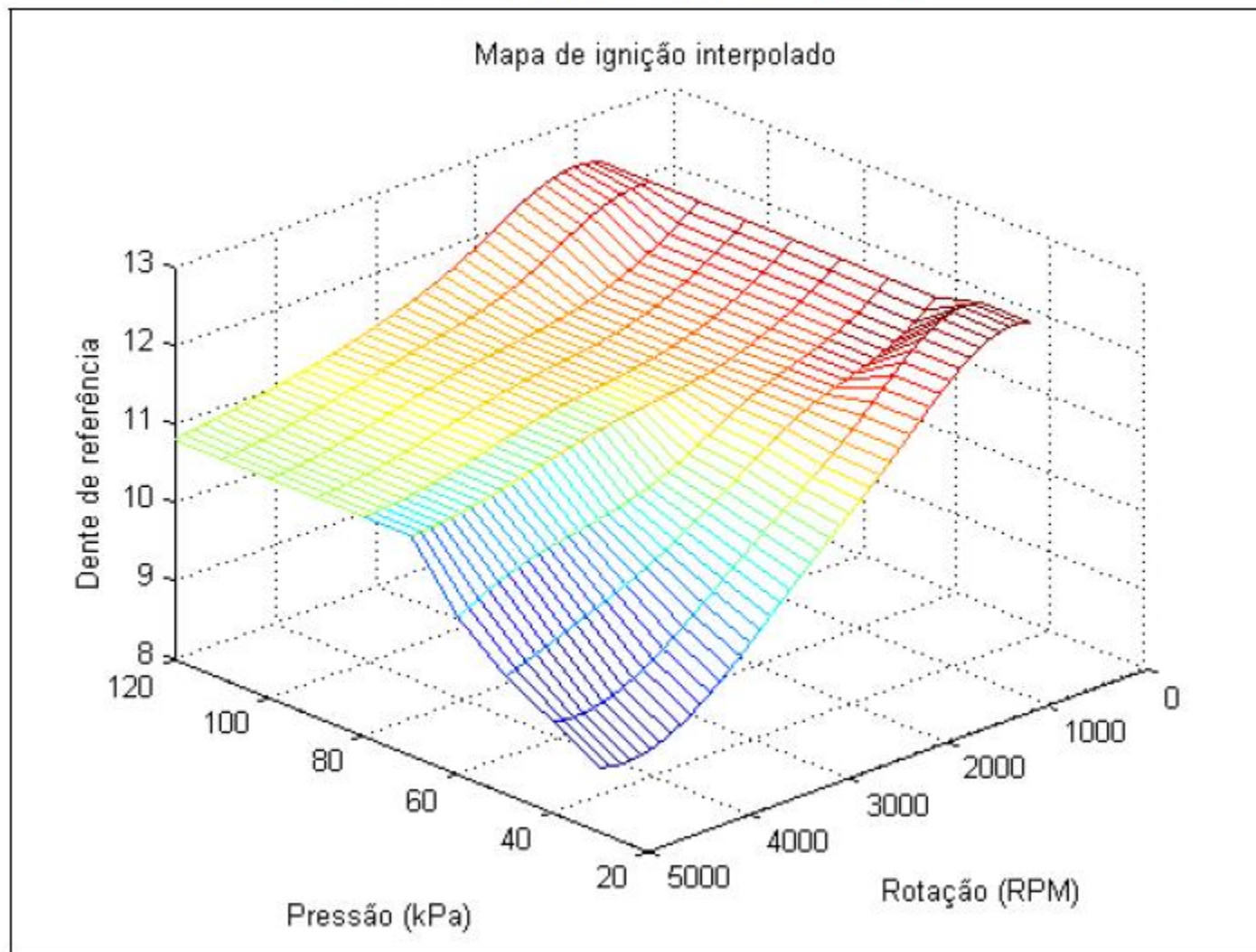
Tabela 5 – Mapa de ignição (em graus) corrigido

RPM kPa	30	40	50	60	70	80	90	100	110	120
500	6	6	7	7	7	7	7	7	7	7
700	9	9	10	10	10	10	10	10	10	10
1300	10	10	13	13	13	13	13	13	11	10
2000	15	15	15	15	15	15	15	15	14	13
3000	22	21	20	19	16	16	16	16	16	16
4000	29	27	24	21	18	18	18	18	18	18
5000	31	29	27	24	19	19	19	19	19	19
6000	31	29	27	24	19	19	19	19	19	19

Tabela 6 – Mapa de ignição final (em posição de dente)

RPM kPa	30	40	50	60	70	80	90	100	110	120
100	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50
200	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50
300	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50	14.50
400	14.10	14.10	14.10	14.10	14.10	14.10	14.10	14.10	14.10	14.10
500	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10
600	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10	13.10
700	12.50	12.50	12.33	12.33	12.33	12.33	12.33	12.33	12.33	12.33
800	12.53	12.54	12.21	12.21	12.23	12.23	12.23	12.23	12.34	12.38
900	12.54	12.55	12.11	12.11	12.14	12.14	12.14	12.14	12.34	12.41
1000	12.52	12.53	12.02	12.03	12.05	12.05	12.05	12.05	12.31	12.42
1100	12.47	12.48	11.95	11.95	11.97	11.97	11.97	11.97	12.28	12.40
1200	12.41	12.41	11.88	11.89	11.90	11.90	11.90	11.90	12.23	12.37
1300	12.33	12.33	11.83	11.83	11.83	11.83	11.83	11.83	12.17	12.33
1400	12.23	12.23	11.78	11.78	11.77	11.77	11.77	11.77	12.10	12.27
1500	12.13	12.12	11.74	11.73	11.71	11.71	11.71	11.71	12.03	12.21
1600	12.01	12.00	11.69	11.69	11.66	11.66	11.66	11.66	11.96	12.14
1700	11.88	11.88	11.65	11.65	11.61	11.61	11.61	11.61	11.88	12.06
1800	11.76	11.75	11.61	11.60	11.57	11.57	11.57	11.57	11.81	11.98
1900	11.63	11.62	11.56	11.55	11.53	11.53	11.53	11.53	11.74	11.91
2000	11.50	11.50	11.50	11.50	11.50	11.50	11.50	11.50	11.67	11.83

Figura 62 – Grafico 3D do mapa de ignição final

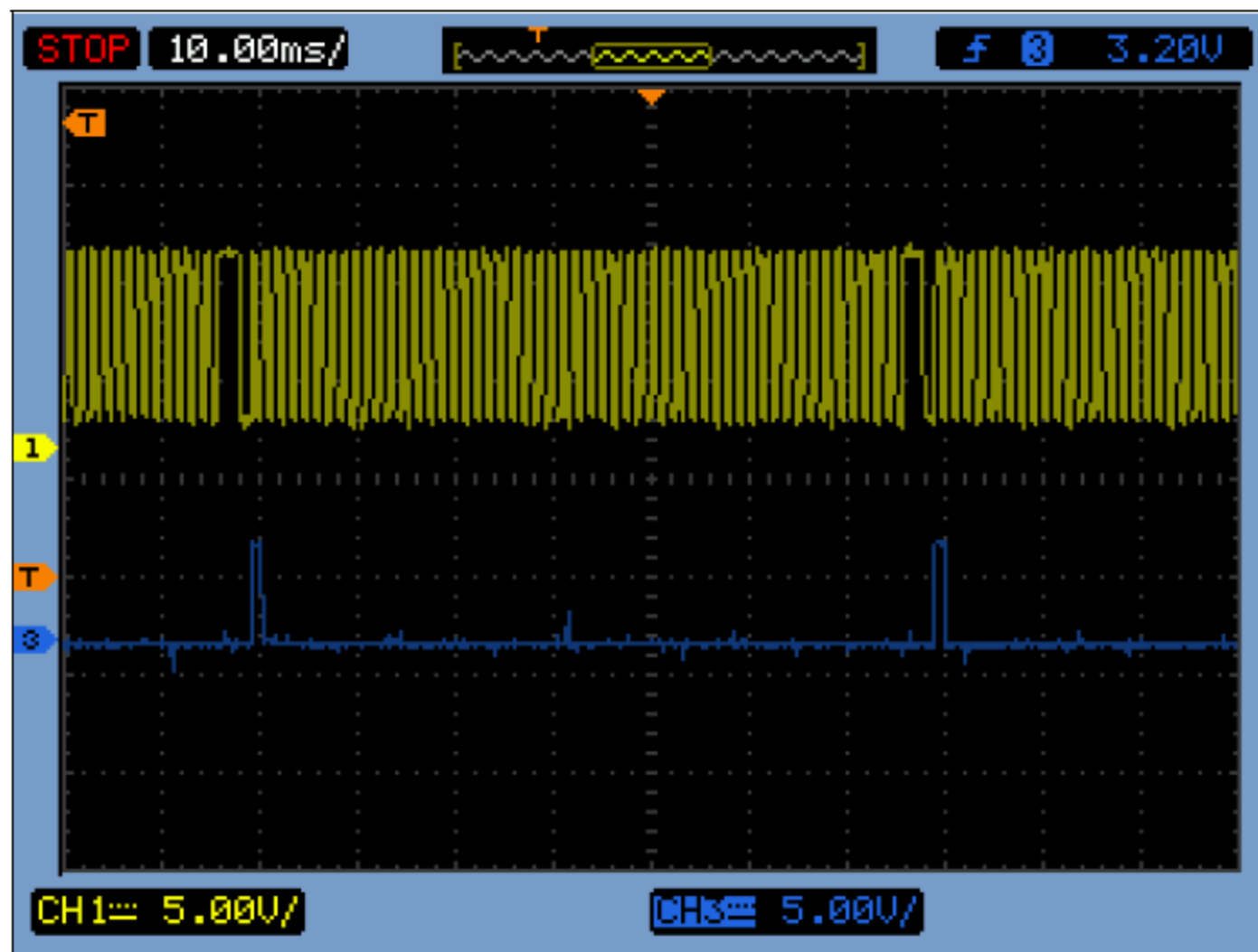


Fonte: o autor

```
// Filtro para impedir aceleracoes bruscas...
if (movimento != 0) {
    if (((ref_rpm_temp - ref_rpm) > 25.0) && (ref_rpm_temp >
        rotacao))
        ref_rpm_temp = ref_rpm + 25.0; //
        Suaviza aceleracao
    else if (((ref_rpm - ref_rpm_temp) > 50.0) && (
        ref_rpm_temp < rotacao))
        ref_rpm_temp = ref_rpm - 50.0; //
        Suaviza desaceleracao
}

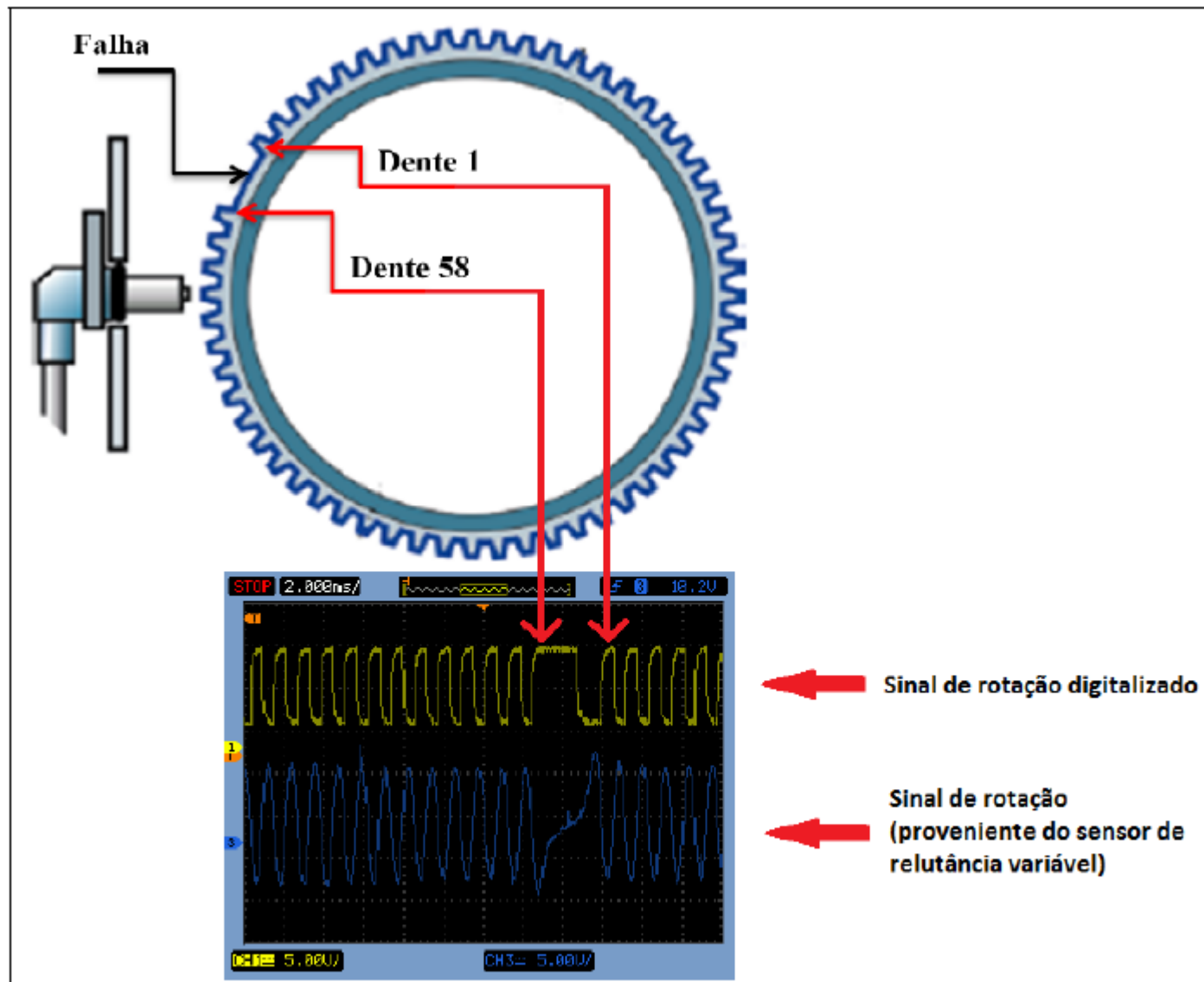
ref_rpm = ref_rpm_temp; // Passa p/ a variavel final
```


Figura 63 – Sinal de rotação e sinal de detecção da falha da roda fônica



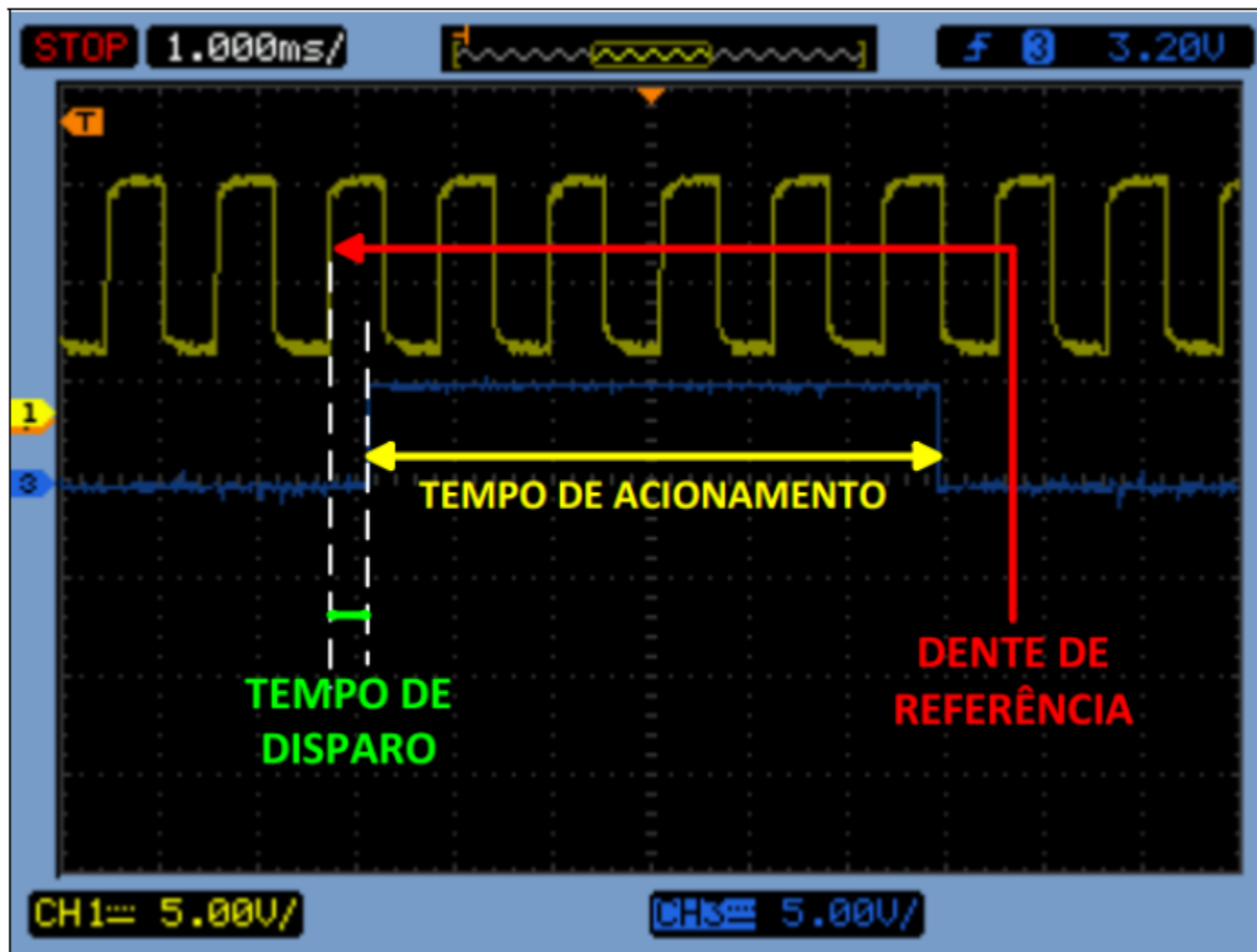
Fonte: o autor

Figura 64 – Relação da roda fônica com o sinal de rotação



Fonte: adaptado de Albaladejo (2013)

Figura 65 – Sinal de ignição com ilustração dos parâmetros de acionamento



```
void conta_dente() {
    dente++; // Contagem das dentes

    ...

    if ((dente == ig_dente_14) && (ig14_pos == 0)) {
        if (ig_tmr_disparo > 9) {
            PITLDO = ig_tmr_disparo;
            PITFLT_PFLTO = 1;
            PITTF = (1<<0); // Limpa flag PTF0
            PITINTE_PINTEO = 1; // Habilita interrupcao timer 0
            ig14_pos = 14;
        }
    }
}
```

```

    }
}

if ((dente == ig_dente_23) && (ig23_pos == 0)) {
    if (ig_tmr_disparo > 9) {
        PITLD3 = ig_tmr_disparo;
        PITFLT_PFLT3 = 1;
        PITTF = (1<<3); // Limpa flag PTF3
        PITINTE_PINTE3 = 1; // Habilita interrupcao timer 3
        ig23_pos = 23;
    }
}

// Se nao ident. a fase uma vez nao deixa dar os sinais de inj
if (rotacao > 6500 || fase_ok == 0) // O GER corta a inj, mas esta
    condicao reforca o corte (seguranca)
    return;

if ((dente == ij_dente_14) && (!volta14) && (ij14_pos == 0)) {
    if (ij_tmr_disparo > 9) {
        PITLD1 = ij_tmr_disparo;
        PITFLT_PFLT1 = 1;
        PITTF = (1<<1); // Limpa flag PTF1
        PITINTE_PINTE1 = 1; // Habilita interrupcao timer 1
        ij14_pos = 1;
    }
}
}

```

```
if ((dente == ij_dente_23) && (volta23) && (ij23_pos == 0)) {
    if (ij_tmr_disparo > 9) {
        PITLD2 = ij_tmr_disparo;
        PITFLT_PFLT2 = 1;
        PITTF = (1<<2); // Limpa flag PTF2
        PITINTE_PINTE2 = 1; // Habilita interrupcao timer 2
        ij23_pos = 2;
    }
}

if ((dente == ij_dente_23) && (!volta23) && (ij23_pos == 0)) {
    if (ij_tmr_disparo > 9) {
        PITLD2 = ij_tmr_disparo;
        PITFLT_PFLT2 = 1;
        PITTF = (1<<2); // Limpa flag PTF2
        PITINTE_PINTE2 = 1; // Habilita interrupcao timer 2
        ij23_pos = 3;
    }
}
```

```
if ((dente == ij_dente_14) && (volta14) && (ij14_pos == 0)) {
    if (ij_tmr_disparo > 9) {
        PITLD1 = ij_tmr_disparo;
        PITFLT_PFLT1 = 1;
        PITTF = (1<<1); // Limpa flag PTF1
        PITINTE_PINTE1 = 1; // Habilita interrupcao timer 1
        ij14_pos = 4;
    }
}
erro = erro & 0b11111101;
} // void conta_dente()
```

```

__interrupt void TIMER_INJ14_PIT1(void)
{
    PITINTE_PINTE1 = 0; // Desabilita int timer 1

    /* Pulsos de injecao p sequencial*/
    switch(ij14_pos) {

        case 0:
            break;

        case 10:
            SINAL_IJ1 = 0; /* Baixa o Sinal da Injecao 1 */

            if (rotacao >= 600 && rotacao_ant >= 600) {

```

³⁴ Um banco é composto pelos cilindros 1 e 4, e o outro, pelos cilindros 2 e 3.


```
volta14 = ~volta14;

// Verifica a logica da fase...
if (rotacao < 2000) { // OBS: Lembrar que a falha eh
    detectada no primeiro dente!
    // Neste caso o sinal de injecao desce DEPOIS da
    falha
    if (sinal_fase == 0) // a fase estava errada!!!
        volta14 = ~volta14;
}
else {
    // Neste caso o sinal de injecao desce ANTES da
    falha
    if (sinal_fase == 1) // a fase estava errada!!!
        volta14 = ~volta14;
}
}

ij14_pos = 0;
break;
```

```

case 40:
    SINAL_IJ4 = 0; /* Baixa o Sinal da Injecao 4 */

    if (rotacao >= 600 && rotacao_ant >= 600)
        volta14 = ~volta14;

    ij14_pos = 0;
    break;

case 1:
    if (tempo_inj > 50) {

        SINAL_IJ1 = 1; /* Ativa Sinal da injecao 1 */

        /* Conta e entao baixa o sinal de injecao por
           interrupcao */
        PITLD1 = tempo_inj;
        PITFLT_PFLT1 = 1;
        PITTF = (1<<1); // Limpa flag PTF1
        PITINTE_PINTE1 = 1; // Habilita interrupcao timer 1

        ij14_pos = 10; // Proxima interrupcao vai desligar o
            sinal
    }
    else
        ij14_pos = 0;

```

```

        break;

    case 4:
        if (tempo_inj > 50) {

            SINAL_IJ4 = 1; /* Ativa Sinal da injeção 4 */

            /* Conta e então baixa o sinal de injeção por
               interrupção */
            PITLD1 = tempo_inj;
            PITFLT_PFLT1 = 1;
            PITTF = (1<<1); // Limpa flag PTF1
            PITINTE_PINTE1 = 1; // Habilita interrupção timer 1

            ij14_pos = 40; // Próxima interrupção vai desligar o
                sinal
        }
        else
            ij14_pos = 0;
        break;
    }
    erro = erro & 0b11101111;
}

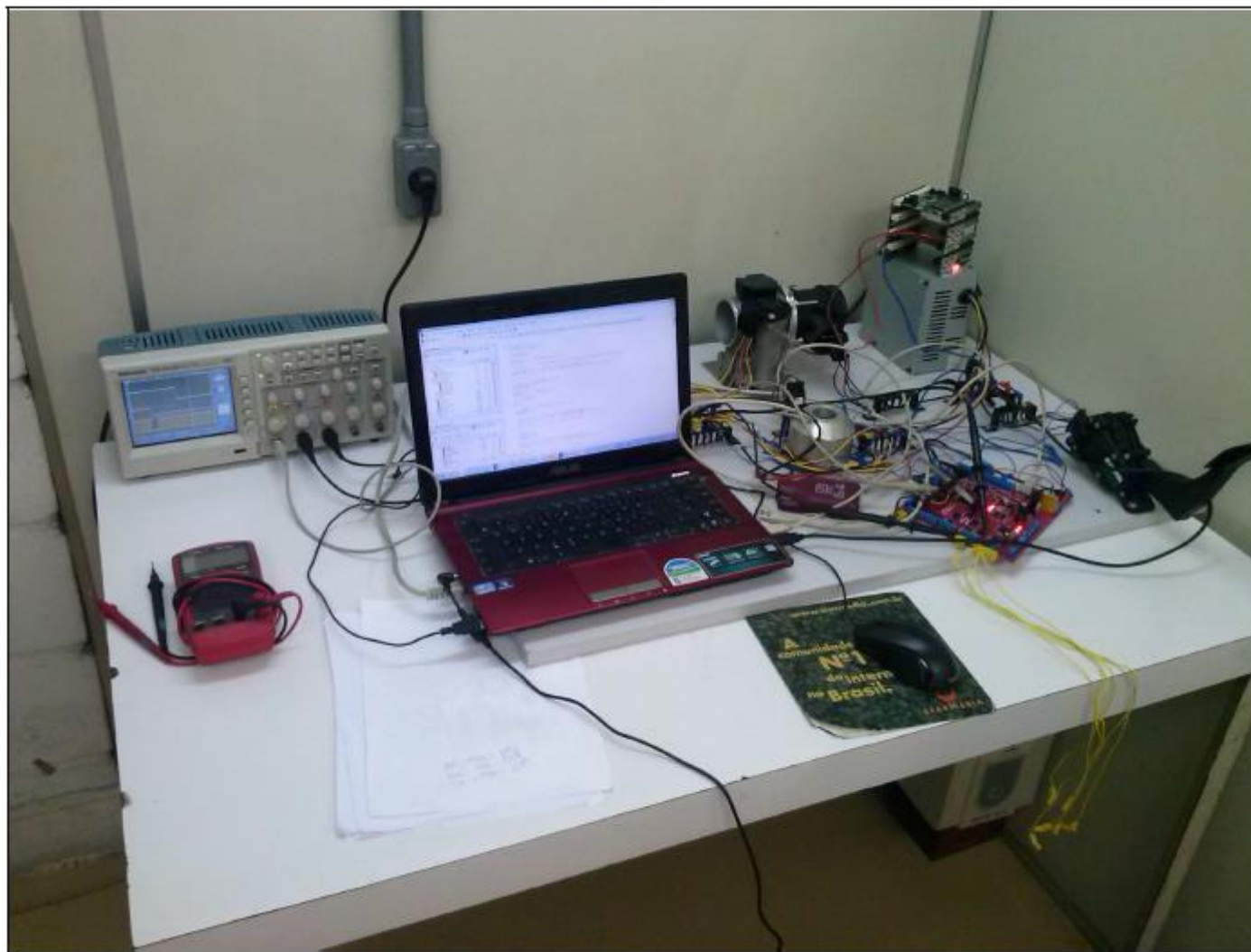
```

Figura 67 – Bancadas com kits de simulação



Fonte: SCARPINETTI e SOARES (2012)

Figura 69 – Laboratório para Testes (LSI-USP)



Fonte: o autor