

# **PROTÓTIPO DE UM COMPUTADOR DE BORDO PARA AUTOMÓVEIS BASEADO NA ARQUITETURA ARM**



**Rafael de Souza**  
**Prof. Miguel Wisintainer - Orientador**

# Roteiro da Apresentação

- **Introdução**
- **Objetivos do trabalho**
- **Microcontrolador ARM**
- **Computador de bordo**
- **Resultados e discussão**
- **Conclusão**
- **Extensões**

# Introdução

- **Computadores de bordo são muito utilizados na Europa**
- **Veículos populares nacionais normalmente possuem poucos mostradores como hodômetro e velocímetro**
- **Funciona por meio de sensores eletrônicos**

# Objetivos do Trabalho

- **Construir a interface entre os diversos sensores presentes no veículo utilizando o microcontrolador ARM LPC2294**
- **Permitir controle de velocidade, emissão de alertas e registro de excessos de velocidade, RPM e temperatura**
- **Disponibilizar quatro hodômetros parciais contendo velocidade média, velocidade máxima, consumo médio, número máximo de RPM alcançado e tempo decorrido do veículo em movimento, ficando os dados armazenados até o reinício do hodômetro**
- **Usar porta serial para configuração do sistema, informando dados dos sensores a serem utilizados**
- **Mostrar os dados através de um LCD**

# Conceitos Básicos

- ▣ **Microcontrolador ARM**
- ▣ **Computador de bordo**
- ▣ **Sensores automotivos**

# Microcontrolador ARM

## Microcontrolador ARM LPC22XX

- Processador de 16/32 bits
- Memória interna de 256 kbytes
- Memória RAM de 16 kbytes
- Velocidade de até 60 MHz
- Possui 144 pinos, sendo 122 de I/O
  - Múltiplas funções
  - 4 grupos de pinos
- Possui 8 canais A/D de 10 bits
- Tensão dos pinos de I/O entre 0 e 3,3 volts



# Computador de bordo

## Conceitos básicos

- ▣ **Velocidade**
- ▣ **Distância percorrida**
- ▣ **Rotações por minuto do motor**
- ▣ **Nível de combustível**
- ▣ **Temperatura do sistema de arrefecimento do motor**
- ▣ **Carga de bateria**

# Sensores automotivos

Sensor	Tipo	Função
Velocidade	Pulsante - digital	Velocidade do veículo
RPM	Pulsante - digital	Quantidade de rotações do motor
Temperatura	Tensão - analógico	Temperatura do motor
Combustível	Tensão - analógico	Nível de combustível no tanque
Bateria	Tensão - analógico	Tensão da bateria do veículo

# Trabalhos Correlatos

## ■ Cristiano Freese

- Computador de bordo utilizando microcontrolador integrado com PC

## ■ David M. Risnik

- Computador de bordo utilizando microcontrolador
- Contador de giros do motor (RPM) utilizando microcontrolador

# Requisitos Principais

## Hardware

- **Coletar o sinal analógico do sensor de temperatura do sistema de arrefecimento**
- **Coletar o sinal analógico do sensor de combustível**
- **Coletar o sinal analógico da tensão da bateria**
- **Coletar o sinal digital dos pulsos do sensor de velocidade**

# Requisitos Principais

## Hardware

- **Coletar o sinal digital dos pulsos das rotações por minuto do motor**
- **Estabelecer comunicação com o LCD**
- **Armazenar os dados na memória EEPROM**
- **Ser implementado utilizando o microcontrolador ARM**

# Requisitos Principais

## Software

- **Permitir a configuração dos sensores através de um PC interligado pela porta serial**
- **Ser implementado em linguagem de programação C, utilizando o ambiente de desenvolvimento WinArm**

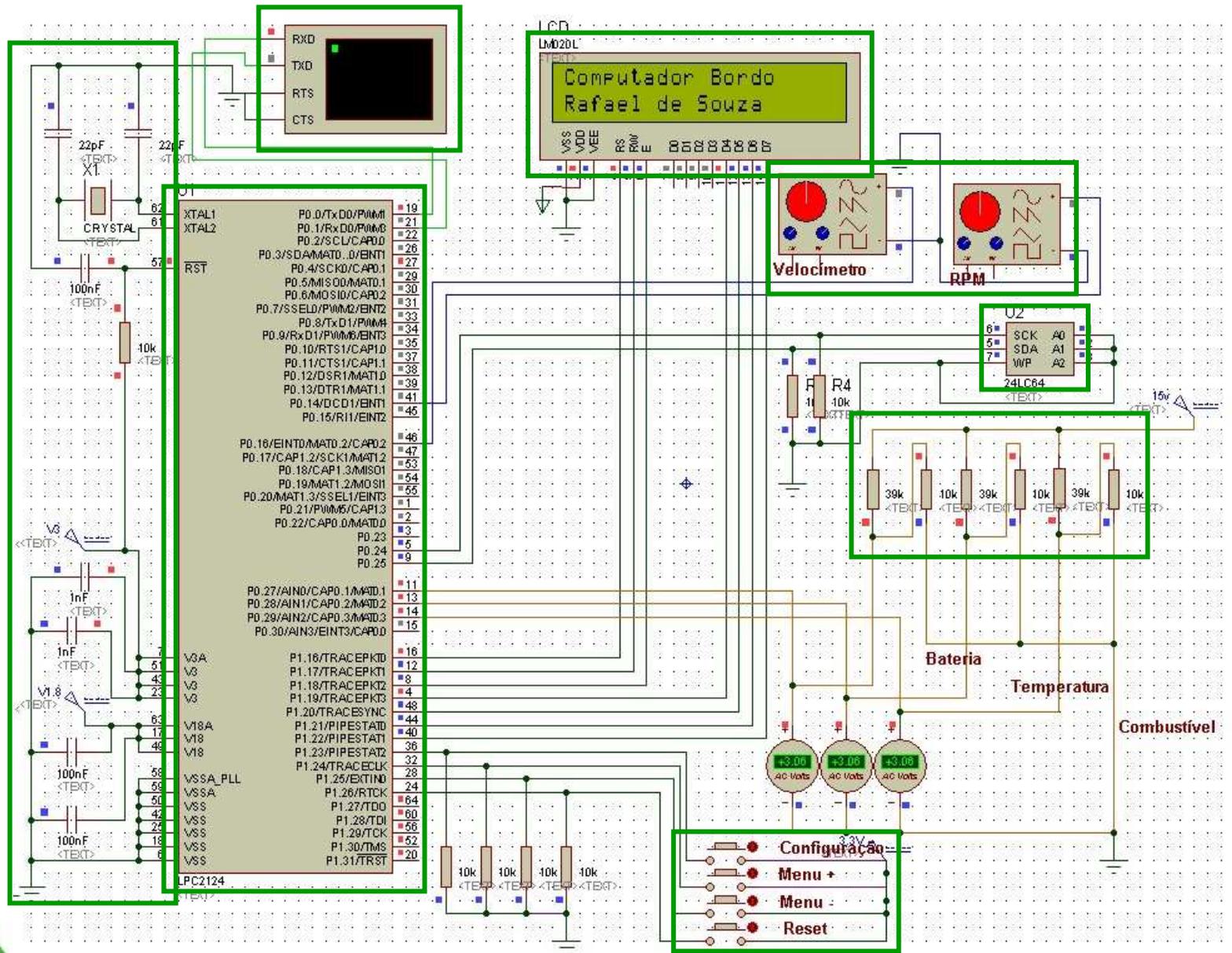
# Especificação

- **Hardware**

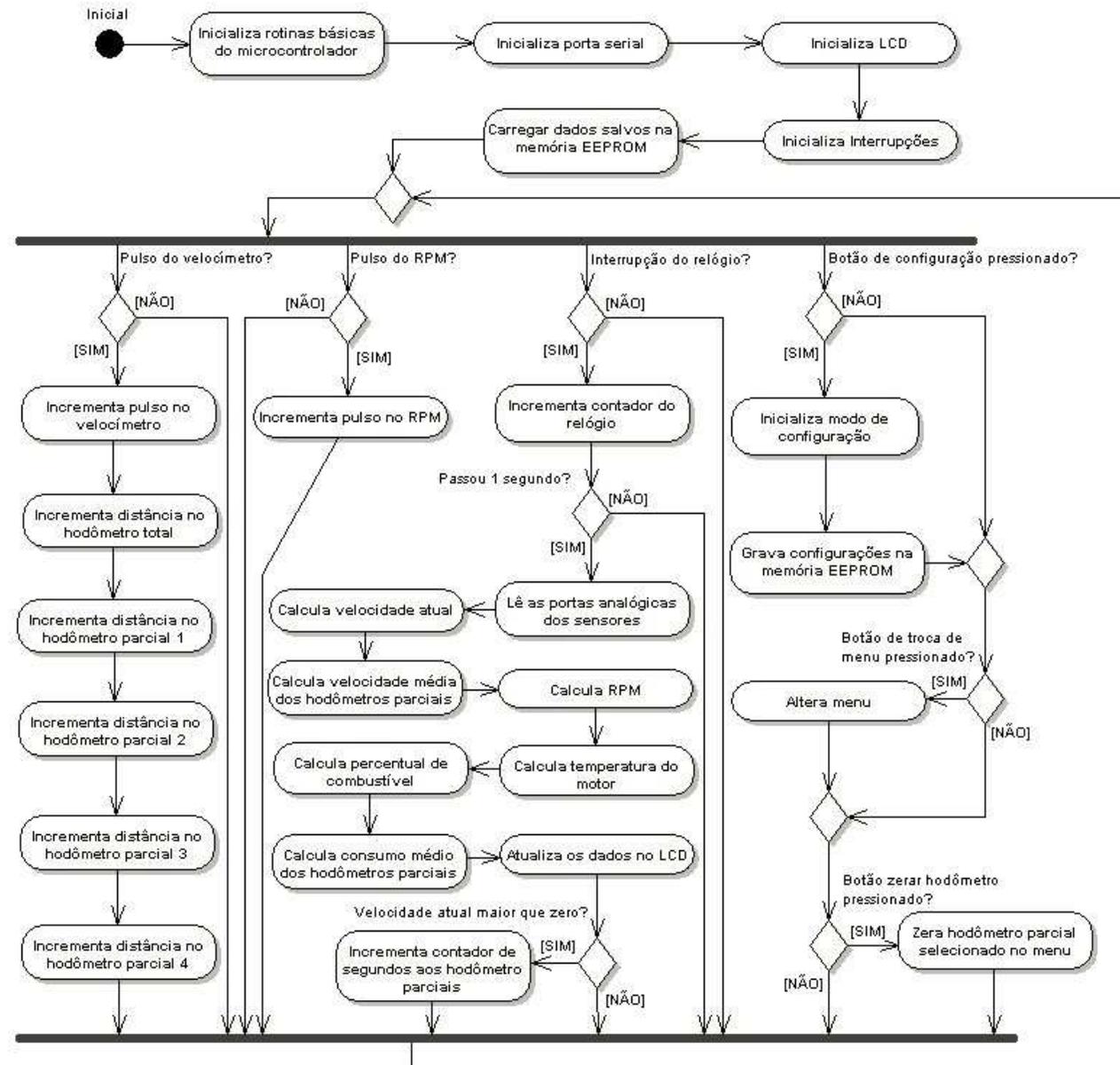
- **Software**

- **Software (configuração)**

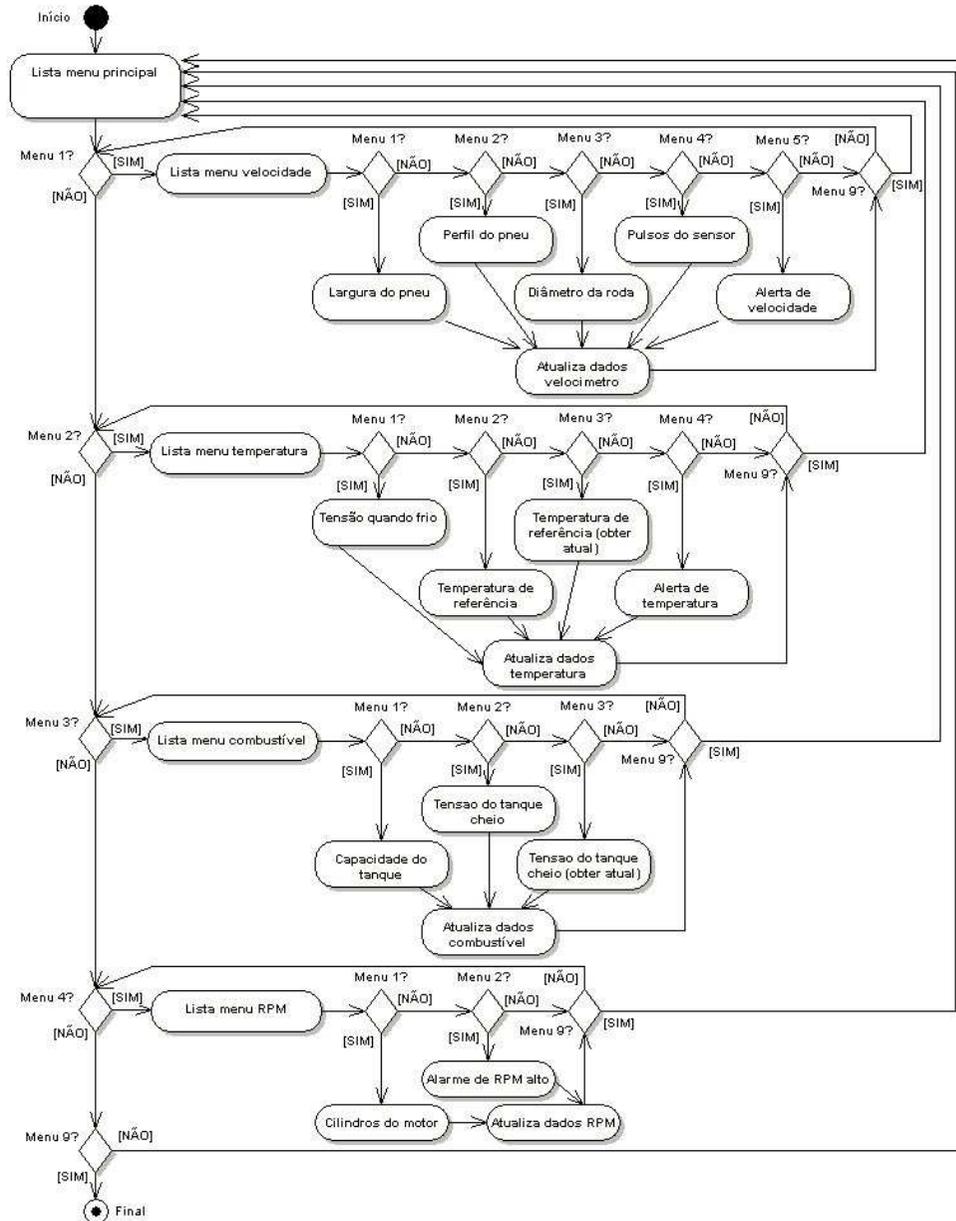
# Especificação Hardware



# Especificação Software



# Especificação Software



# Ferramentas Utilizadas

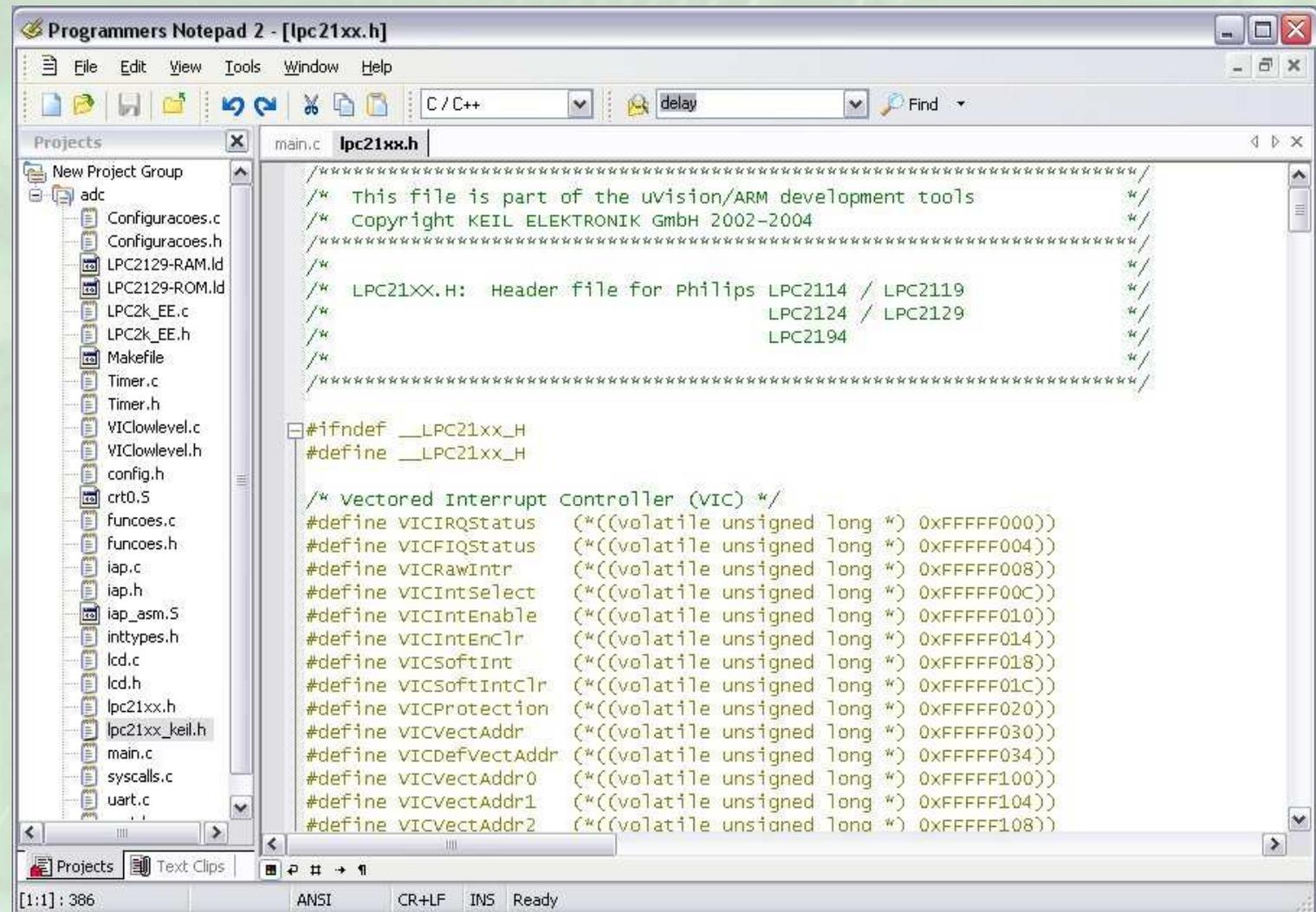
- ▣ **Hyperterminal**
- ▣ **Proteus Professional**
- ▣ **WinArm**
- ▣ **LPC2000 Flash Utility**
- ▣ **Enterprise Architect**

# Implementação

- **Ambiente WinArm**
- **LPC2000 Flash Utility**
- **Simulação Proteus Professional**
- **Implementação no Protoboard**

# Técnicas e Ferramentas Utilizadas

## Ambiente WinArm

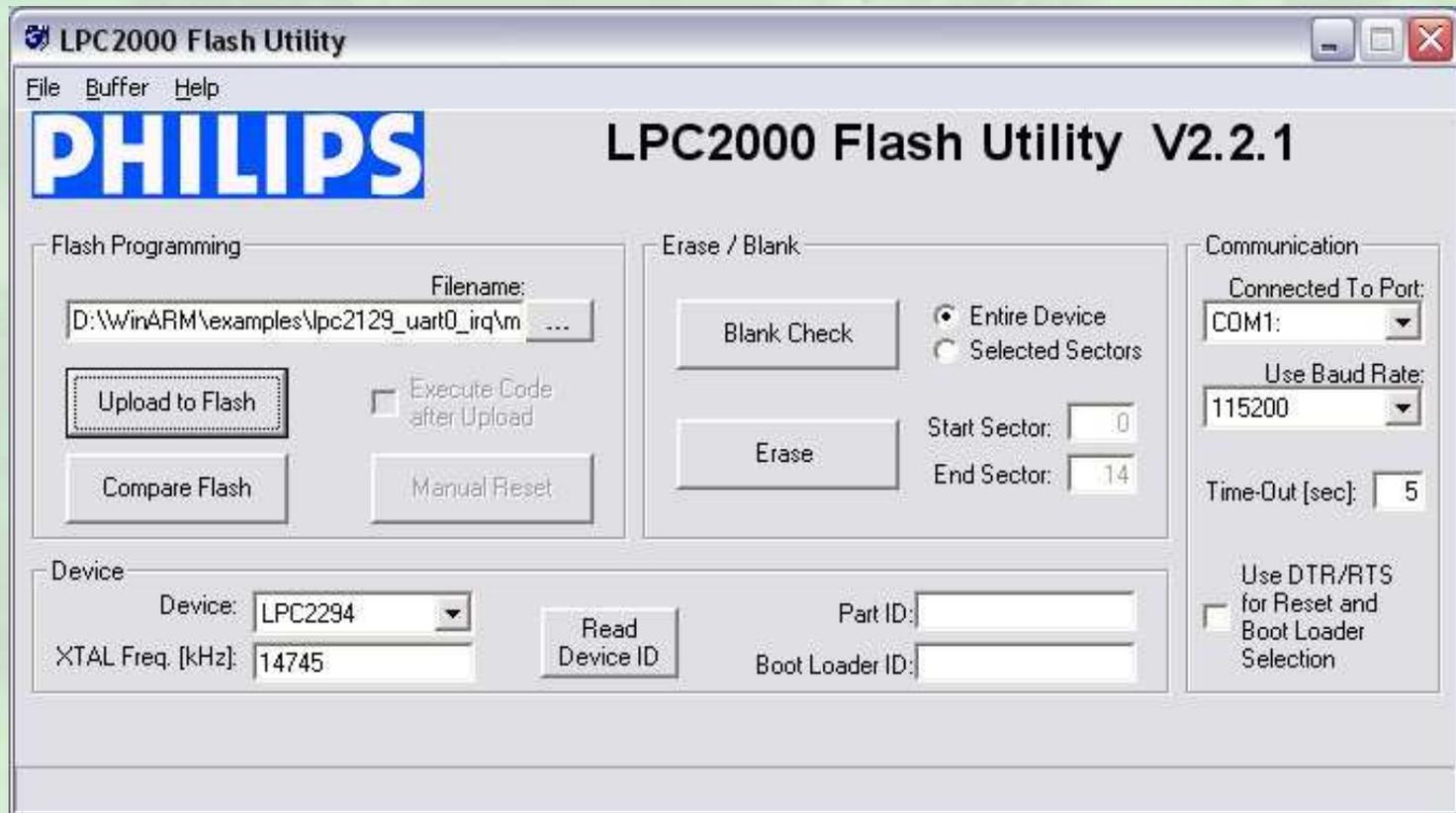


```
Programmers Notepad 2 - [lpc21xx.h]
File Edit View Tools Window Help
C / C++ delay Find
Projects
New Project Group
adc
  Configuracoes.c
  Configuracoes.h
  LPC2129-RAM.ld
  LPC2129-ROM.ld
  LPC2k_EE.c
  LPC2k_EE.h
  Makefile
  Timer.c
  Timer.h
  VIClowlevel.c
  VIClowlevel.h
  config.h
  crt0.S
  funcoes.c
  funcoes.h
  iap.c
  iap.h
  iap_asm.S
  inttypes.h
  lcd.c
  lcd.h
  lpc21xx.h
  lpc21xx_keil.h
  main.c
  syscalls.c
  uart.c
main.c lpc21xx.h
/*****
/* This file is part of the uvision/ARM development tools
/* Copyright KEIL ELEKTRONIK GmbH 2002-2004
*****/
/*
/* LPC21XX.H: Header file for Philips LPC2114 / LPC2119
/* LPC2124 / LPC2129
/* LPC2194
*****/
/*****
#ifndef __LPC21XX_H
#define __LPC21XX_H

/* vectored Interrupt Controller (VIC) */
#define VICIRQStatus (*(volatile unsigned long *) 0xFFFFF000)
#define VICFIQStatus (*(volatile unsigned long *) 0xFFFFF004)
#define VICRawIntr (*(volatile unsigned long *) 0xFFFFF008)
#define VICIntSelect (*(volatile unsigned long *) 0xFFFFF00C)
#define VICIntEnable (*(volatile unsigned long *) 0xFFFFF010)
#define VICIntEnClr (*(volatile unsigned long *) 0xFFFFF014)
#define VICSoftInt (*(volatile unsigned long *) 0xFFFFF018)
#define VICSoftIntClr (*(volatile unsigned long *) 0xFFFFF01C)
#define VICProtection (*(volatile unsigned long *) 0xFFFFF020)
#define VICVectAddr (*(volatile unsigned long *) 0xFFFFF030)
#define VICDefVectAddr (*(volatile unsigned long *) 0xFFFFF034)
#define VICVectAddr0 (*(volatile unsigned long *) 0xFFFFF100)
#define VICVectAddr1 (*(volatile unsigned long *) 0xFFFFF104)
#define VICVectAddr2 (*(volatile unsigned long *) 0xFFFFF108)
#endif
```

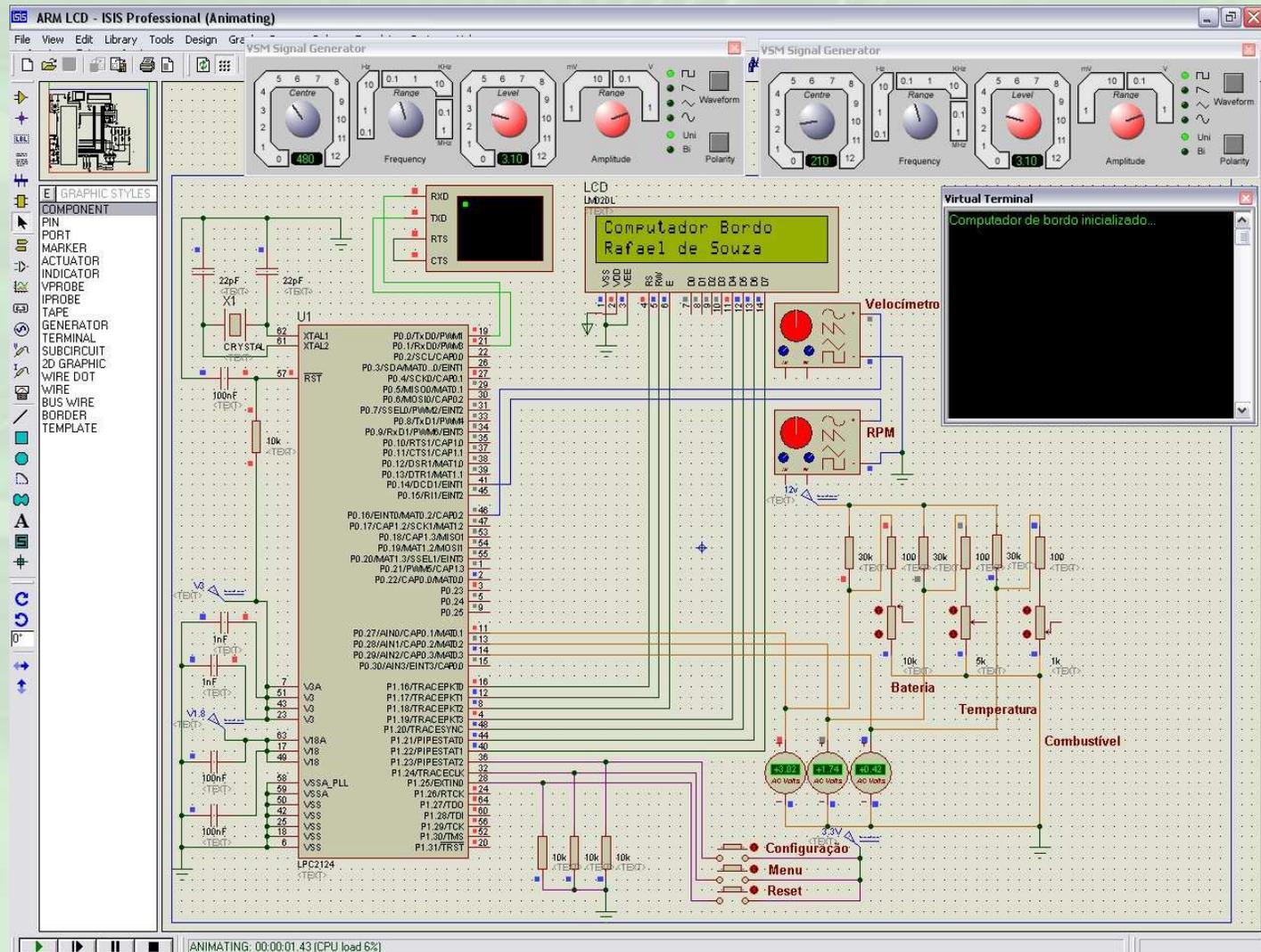
# Técnicas e Ferramentas Utilizadas

## LPC2000 Flash Utility



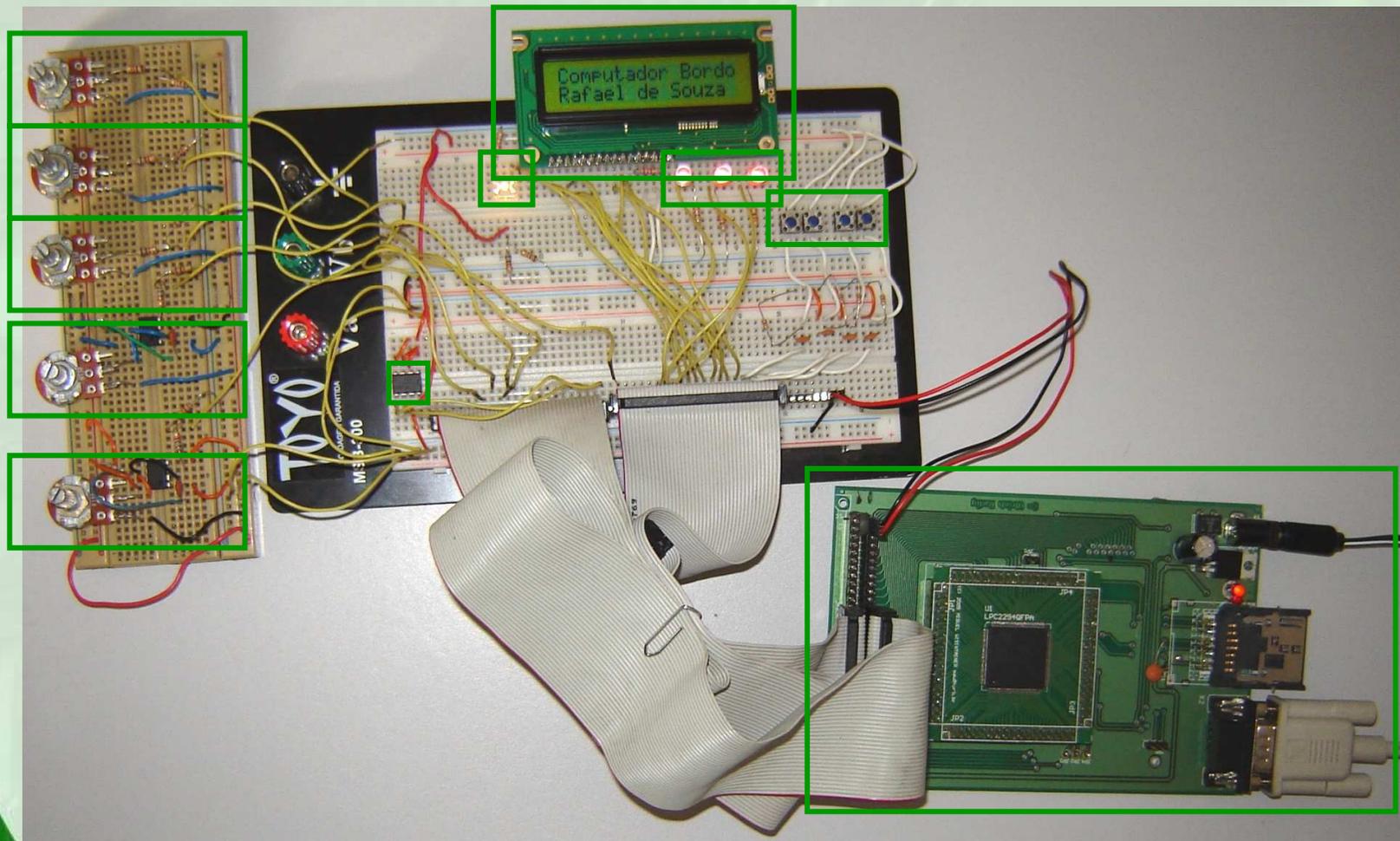
# Técnicas e Ferramentas Utilizadas

## Simulação com o Proteus Professional



# Técnicas e Ferramentas Utilizadas

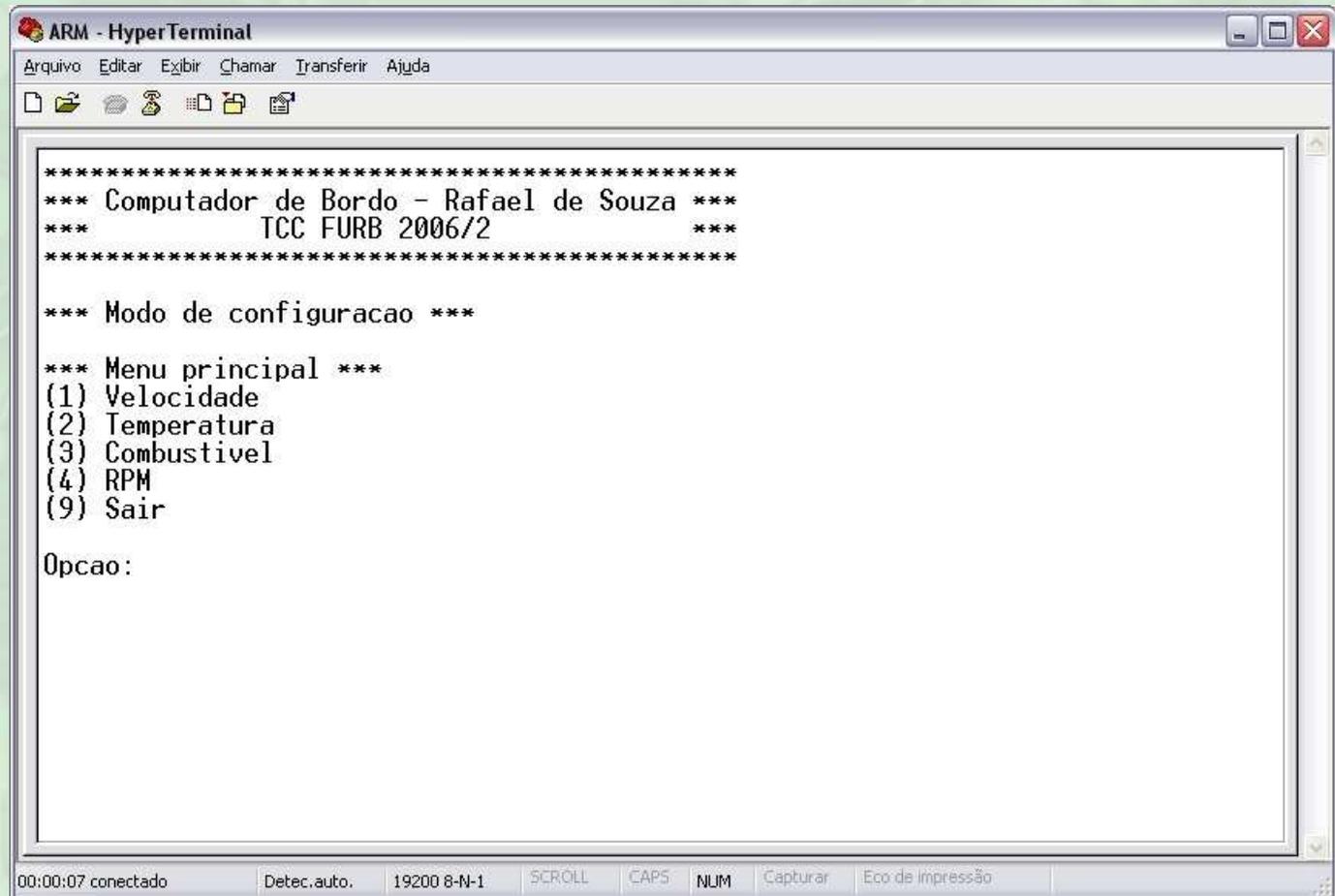
## Implementação no Protoboard



# Estudo de caso

## Modo de configuração

- Menu principal



The image shows a screenshot of a HyperTerminal window titled "ARM - HyperTerminal". The window has a menu bar with "Arquivo", "Editar", "Exibir", "Chamar", "Transferir", and "Ajuda". Below the menu bar is a toolbar with icons for file operations. The main text area displays the following content:

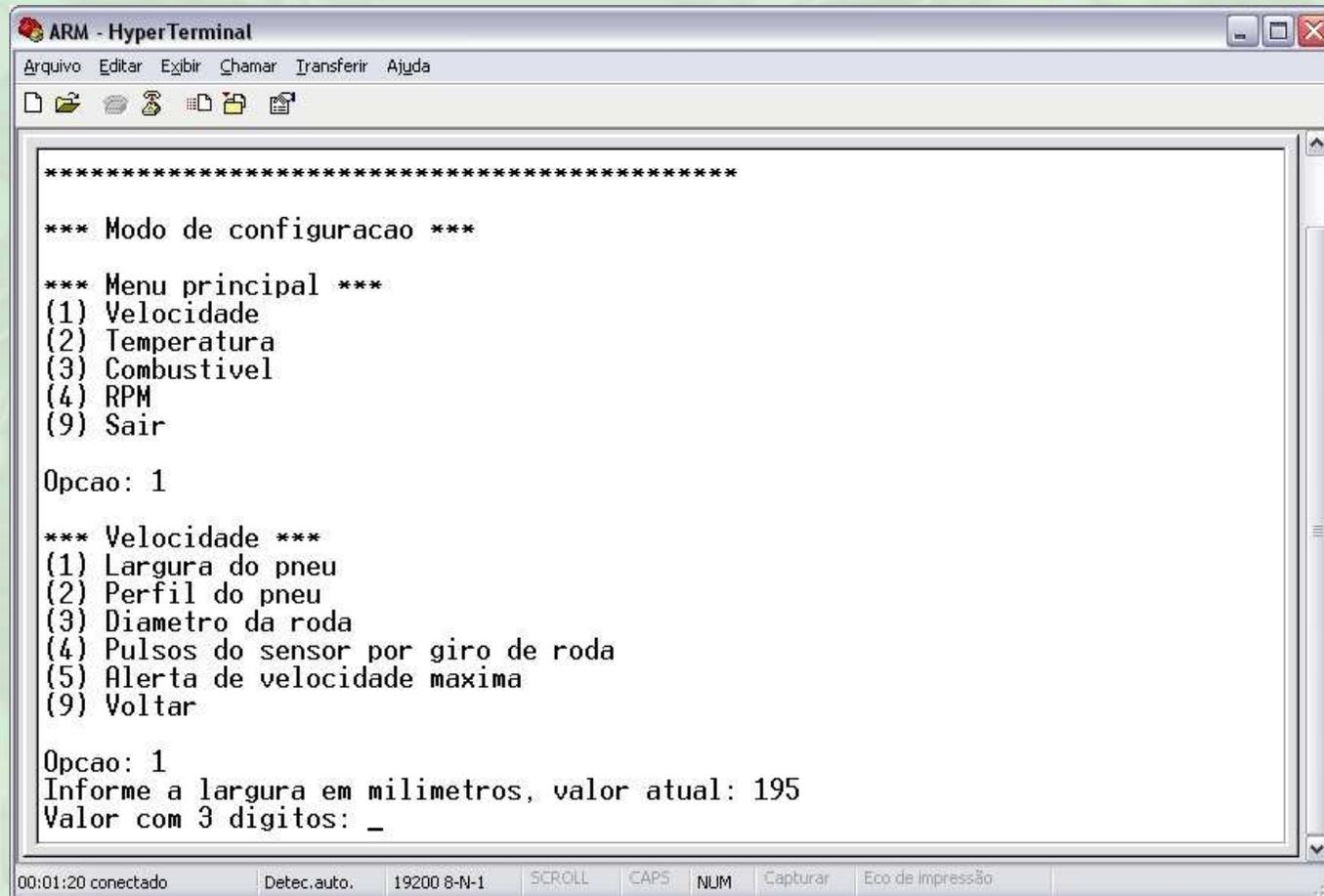
```
*****  
*** Computador de Bordo - Rafael de Souza ***  
***           TCC FURB 2006/2           ***  
*****  
  
*** Modo de configuracao ***  
  
*** Menu principal ***  
(1) Velocidade  
(2) Temperatura  
(3) Combustivel  
(4) RPM  
(9) Sair  
  
Opcao:
```

At the bottom of the window, there is a status bar with the following information: "00:00:07 conectado", "Detec. auto.", "19200 8-N-1", "SCROLL", "CAPS", "NUM", "Capturar", and "Eco de Impressão".

# Estudo de caso

## Modo de configuração

- Alterando configuração



```
ARM - HyperTerminal
Arquivo Editar Exibir Chamar Transferir Ajuda
*****
*** Modo de configuracao ***

*** Menu principal ***
(1) Velocidade
(2) Temperatura
(3) Combustivel
(4) RPM
(9) Sair

Opcao: 1

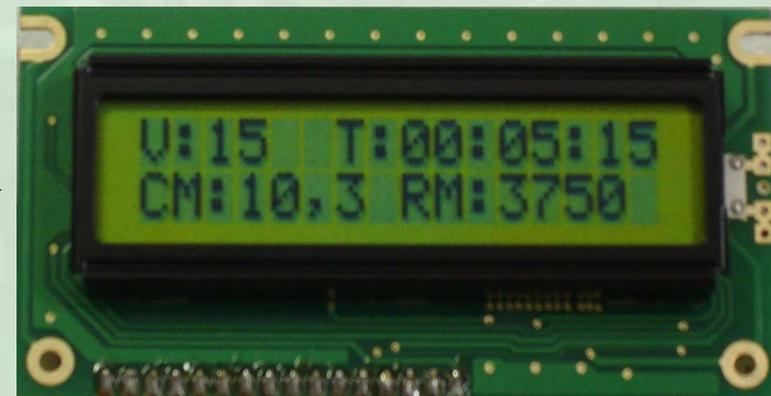
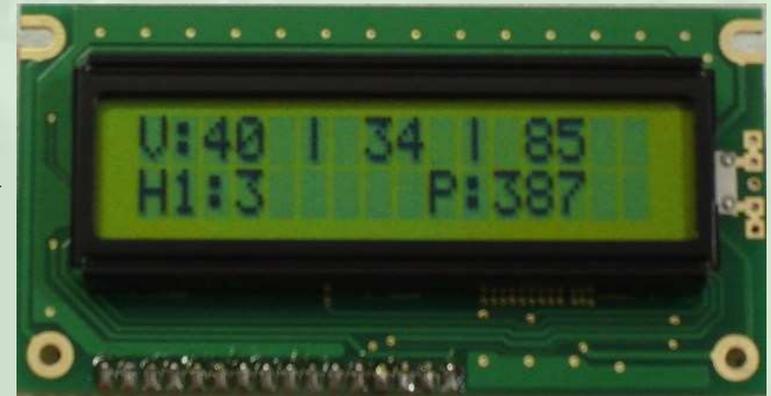
*** Velocidade ***
(1) Largura do pneu
(2) Perfil do pneu
(3) Diametro da roda
(4) Pulsos do sensor por giro de roda
(5) Alerta de velocidade maxima
(9) Voltar

Opcao: 1
Informe a largura em milímetros, valor atual: 195
Valor com 3 digitos: _

00:01:20 conectado   Detec.auto.   19200 8-N-1   SCROLL   CAPS   NUM   Capturar   Eco de Impressão
```

# Estudo de caso

## Execução



# Resultados e Discussão

- ▣ **Apresentou bons resultados realizando corretamente a leitura dos sensores**
- ▣ **Dificuldade em encontrar material para o microcontrolador ARM**
- ▣ **Desenvolvimento de rotinas**
- ▣ **Dificuldade em simular o microcontrolador ARM no Proteus**

# Conclusão

- ▣ **Emprego do computador de bordo em veículos que não o possuem de fábrica**
- ▣ **Emprego de microcontrolador ARM**
- ▣ **O TCC será uma fonte para iniciantes em programação para ARM**

# Extensões

- **Utilizar LCD gráfico**
- **Utilizar sensores de indicação de aproximação**
- **Implementação de sistema inteligente de detecção de fadiga do motorista**
- **Implementar um tocador de músicas**
- **Implementar um sistema de GPS**