Keil MSBSTM32 board
PA
  PA0 WKUP switch input
  PA1 Analog in from pot
  PA4 SD-Adapter
  PA5 SPI SCLK SD-Adapter
  PA6 SPI MISO SD-Adapter
  PA7 SPI MOSI SD-Adapter
  PA8 SD-Adapter
  PA9 USART1 Tx RS232 out
  PA10 USART1 Rx RS232 in
  PA11 USB DM
  PA12 USB DP
  PA13 JTAG
  PA14 JTAG
  PA15 JTAG

PB
  PB2 Boot1
  PB3 JTAG
  PB4 JTAG
  PB8 LED CAN Rx in
  PB9 LED CAN Tx out
  PB10 LED
  PB11 LED
  PB12 LED
  PB13 LED
  PB14 LED
  PB15 LED

PC
  PC0 DB7 LCD data
  PC1 DB6 LCD data
  PC2 DB5 LCD data
  PC3 DB4 LCD data
  PC10 E LCD clock
  PC11 RW LCD Read=1, write=0
  PC12 RS LCD control/data register select
  PC13 TAMP switch input
  PC14 32.768 crystal in
  PC15 32.768 crystal out

PD
  PD0 OSC_In
  PD1 OSC_Out
  PD2 VUSB

Unused pins (FT= five voltage tolerance input)
PA
  PA0 WKUP switch input, can be used for other input, has a pull-up to 3.3V
  PA2 USART2_TX, ADC2, TIM2_CH2
  PA3 USART2_RX, ADC3, TIM2_CH3

PB
  PB0 ADC8, TIM3_CH3
  PB1 ADC9, TIM3_CH4
  PB5 TIM3_CH2
  PB6 I2C_SCL, TIM4_CH1
  PB7 I2C_SDA, TIM4_CH2
  PB10 LED, I2C_SCL, TIM2_CH3 (FT)
  PB11 LED, I2C_SDA, TIM2_CH4 (FT)
  PB12 LED, SPI2_NSS, TIM1_BKIN (FT)
  PB13 LED, SPI2_SCK, TIM1_CH1N (FT)
  PB14 LED, SPI2_MISO, TIM1_CH2N (FT)
  PB15 LED, SPI2_MOSI, TIM1_CH3N (FT)

PC
  PC4 ADC14
  PC5 ADC15
  PC6 TIM3_CH1 (FT)
  PC7 TIM3_CH2 (FT)
  PC8 TIM3_CH3 (FT)
  PC9 TIM3_CH4 (FT)
  PC13 TAMP switch input, can be used for other input, has a pull-up to 3.3V

Rough plan Labs 1-6 in teams of 2
Lab 1) Run various starter programs and asking student to find information in the data sheets and example code. Convert LCD and interrupting serial port code into driver code. Include time of day clock. Write an interpreter.

Lab 2) Design, implement, and test a thread switching kernel with sleeping, blocking semaphores, pipes and priority using Lab 1. Schedule both periodic tasks in the background as ISRs and foreground tasks using a thread switcher. Inputs from serial port interpreter, switches and the ADC connected to potentiometer. Outputs to serial port interpreter, LEDs and LCD.

Lab 3) Add second periodic interrupt, second button task, blocking semaphores and priority scheduler. While running the RTOS, output debugging data into PC for plotting and storage. Record when, where and what the real time system is doing.

Lab 4) Sample audio, run FFT and plot results on LCD.

Lab 5) Interface the IR distance sensor from the robot kit with no filtering. Design, implement, and test a file system with existing hardware on board. Record distance data or debugging data onto SD card. Dump data via USB to PC. Do not implement all of FAT32.

Lab 6) Add analog and digital filtering (averaging or median) to the IR distance sensors interface. Transmit analog signals across CAN. Most of the low-level CAN code is available.

Lab 7A) Design implement and test a PI motor control of one motor/sensor running on RTOS (add to labs 1,2,3,4)

Lab 7B) Robot competition in teams of 3, 4 or 5. There are 19 unused pins
3 analog inputs from IR sensors
9 digital outputs to 3 DC motors
2 timer inputs from wheel sensors
1 timer input from Ping
1 digital output to servo
3 touch sensors