TBSAR Three-Layer Architecture with Task Scheduler

APPLICATION LAYER

Task Scheduler
SysTick @ 10ms base tick
Files: App/Scheduler.c/h

Task_10ms
Executes every 10ms

Task_20ms
Executes every 20ms

Task_50ms
Executes every 50ms

Task_100ms
RGB LED toggle (Red @ 100ms)

main.c
Application entry point

↕ **SysTick Interrupt (10ms periodic timer)**

ECU LAYER (Feature Modules)

Dac Module
SPI DAC control
Files: Ecu/Dac/

Eeprom Module
I2C external memory
Files: Ecu/Eeprom/

Monitor Module
UART debug printf
uart_printf() @ 57600 baud

Rgb Module
RGB LED control
Rgb_Config(), Rgb_Set()

Sensors Module
ADC + conversions
LM35 temp, VPOT voltage

Ukeys Module
5 User Keys scanning
Files: Ecu/Ukeys/

MCU LAYER (Hardware Drivers)

LPC11C24 MCU
ARM Cortex-M0 @ 48MHz
32KB Flash, 8KB SRAM

Adc Driver
10-bit ADC
Files: Mcu/Adc/

Can Driver
CAN bus
Files: Mcu/Can/

Dio Driver
GPIO control
Files: Mcu/Dio/

Gpt Driver
Timer/PWM driver
Files: Mcu/Gpt/

I2c Driver
I2C @ 400kHz
Files: Mcu/I2c/

Spi Driver
SPI @ 1MHz
Files: Mcu/Spi/

Uart Driver
UART @ 57600 baud
Files: Mcu/Uart/

SysTick
System timer (10ms)
Triggers scheduler

Sys
Startup, Clock, Syscon
Files: Mcu/Sys/

HARDWARE

RGB LED • 5 User Keys • LM35 Temperature Sensor • I2C EEPROM • SPI DAC

Current Application

Task Scheduler with RGB Red LED blink @ 100ms

Build: 7668 bytes Flash (23.6%), 584 bytes RAM (7.1%)
Toolchain: arm-none-eabi-gcc with -O2 optimization
Programming: lpc21isp @ 57600 baud

Architecture Flow:

1. SysTick timer generates 10ms interrupts
2. Scheduler calls tasks at 10/20/50/100ms periods using countdown timers
3. Tasks use ECU modules (Rgb, Sensors, Monitor, etc.)
4. ECU modules call MCU drivers (Dio, Adc, Uart, I2c, Spi, etc.)
5. MCU drivers access hardware registers to control peripherals