CHAPTER 3

HUMAN MOTION DETECTION AND HARDWARE ARCHTECTURE

3.1 BACKGROUND IMAGE PROCESSING APPROACH

For real-time target extraction the background image processing approach (i.e. background subtraction) is used to provide the most complete feature dataset, but it is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Background image is not fixed and it must adapt to

(i) Motion changes like tree branch move

(ii) Changes in background because of objects entering in a scene.

(iii) Stay for longer period without motion.

3.1.1 Feature extraction of human motion detection

The basic method involves modeling background as color image. Background is estimated as the average or the median of the previous ‘n’ frames. If the distance between current frame and background is greater than some threshold for some pixel then that is considered as foreground. The various operations are performed to reduce the noise and find the threshold value in the original color image and detect the human motion features as shown in Figure 3.1.

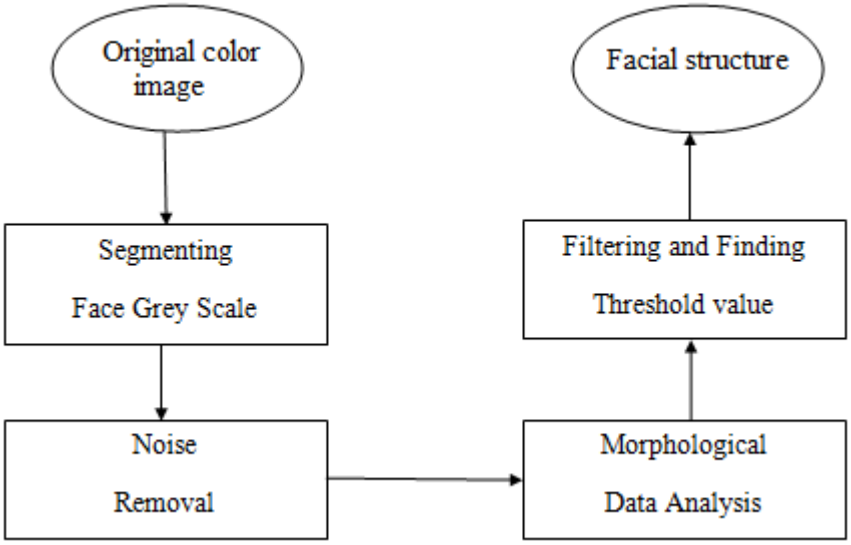


Figure 3.1 Operations on detecting human motion features

In this research, a new method is proposed to detect moving object based on background subtraction and the steps are included as:

- Background initialization
- Background updating model
- Foreground mask
- Post processing and Dynamic threshold

The comparison between existing and proposed system of human motion estimation is tabulated in Table 3.1.

Table 3.1 Comparison between existing and proposed system

EXISTING SYSTEM	PROPOSED SYSTEM
• Optical Flow Detection Method • Frame Separation Method	• Background Subtraction Method
DRAWBACKS	**OVERCOME**
• Large quantity of calculation, sensitivity to noise. • Difficult to obtain a complete outline of moving object.	• Simple algorithm, important role in human body tracking. • Effective to enhance the effect of moving object detection

3.2 BLOCK DIAGRAM OF PROPOSED SYTEM

In this work, a single static camera condition is identified and combined with a dynamic background modeling with dynamic threshold. The human motion can be detect and estimate through the proposed background subtraction algorithm is obtained and the design of the proposed system is shown in Figure 3.2

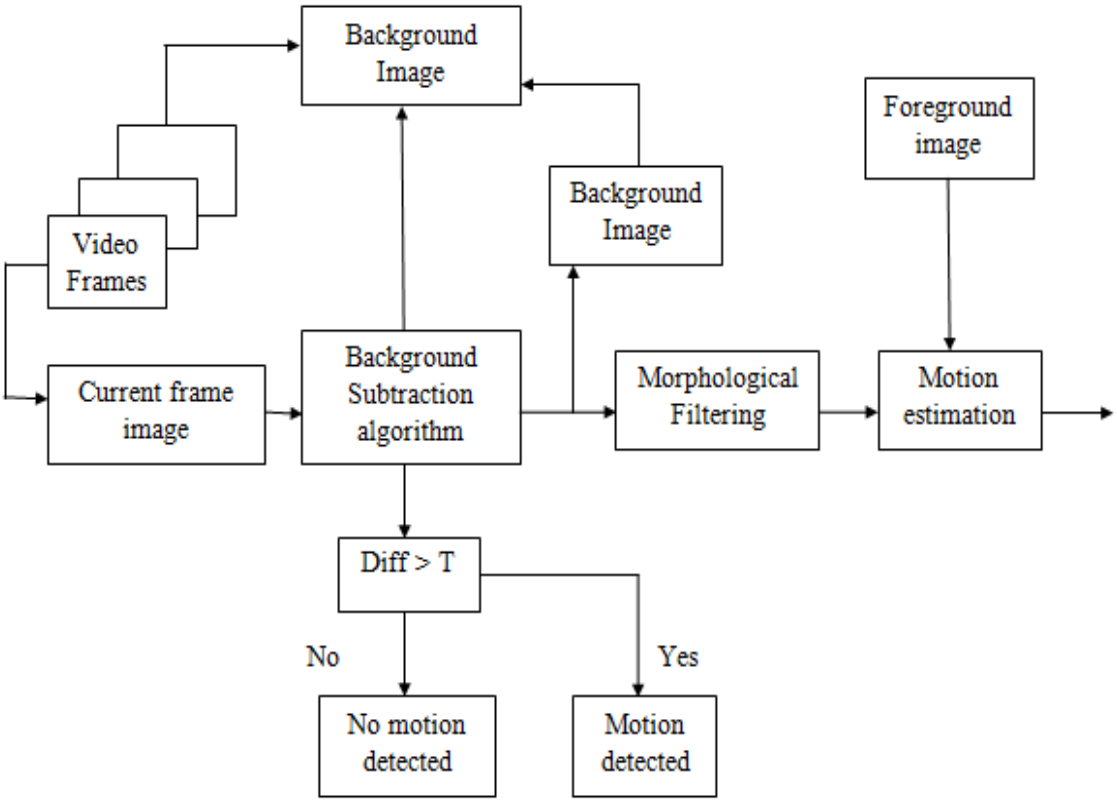


Figure 3.2 Block diagram of proposed system

3.2.1 Background Initialization

The method of taking the median from continuous multiframe can be resolved simply and effectively by using the background algorithm given by

$$B_{ini}(x,y) = \text{median } F_k(x,y) \qquad K = 1,2,3\ldots\ldots n \qquad \ldots (3.1)$$

where B_{ini} is the initial background and 'n' is the total number of frames selected.

3.2.1.1BACKGROUND UPDATE

For the better adapt to light changes, the background needs to be updated in real-time, so as to accurately extract the moving object.

UPDATE ALGORITHM

$$B_{k+1}(x,y) = \beta\, B_K(x,y) + (1-\beta)\, F_K(x,y) \quad \ldots (3.2)$$

where, $F_K(x,y)$ - pixel gray value in the current frame

$B_{k+1}(x,y)$- background value of the next frame

$B_K(x,y)$ - background value of the current frame

β - 0.0004 update coefficient

3.2.2 Background Subtraction

After the background image B(x,y) is obtained , subtract the background image B(x,y) from the current frame.

$$D_k(x,y) = \begin{cases} 1 & |F_k(x,y) - B_{k-1}(x,y)| > T \\ 0 & others \end{cases} \quad \ldots (3.3)$$

where, $D_k(x,y)$ – Binary image of differential results

T – Gray scale threshold

3.3 INFERENCE RESULTS FROM BACKGROUND SUBRACTION

The motion analysis scheme was tested on a database of video sequence of human walking. The pixels on target ranged from approximately 50 to 400. The output of human motion estimation corresponding to background subtraction is obtained and the binary foreground images are processed after the background process as shown in Figure 3.3 (a) to Figure 3.3(d).

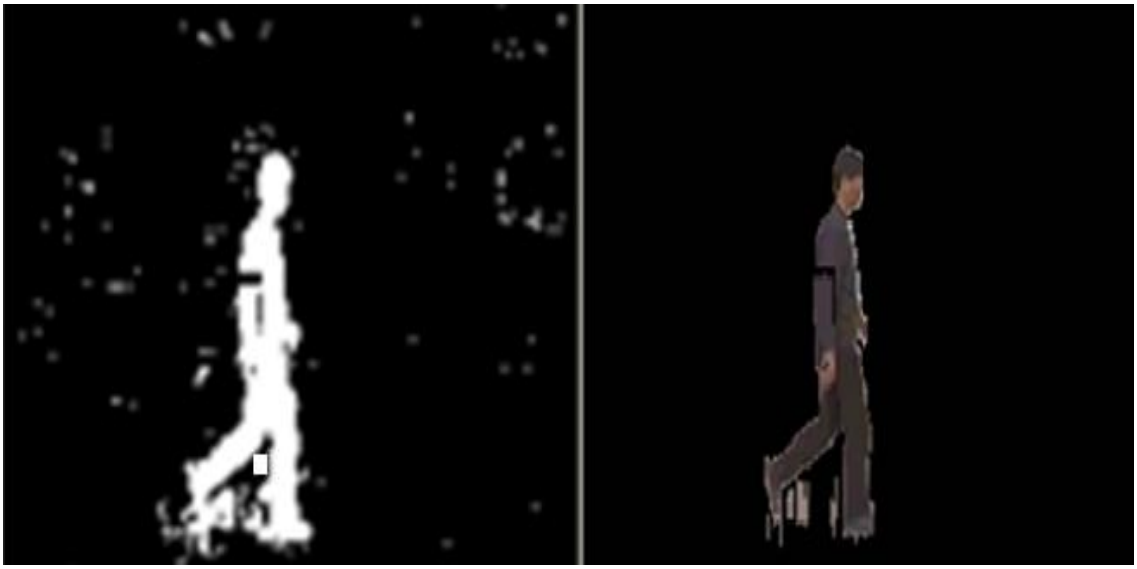
(a) (b)

Figure 3.3(a) Typical scene to be processed. (b) Binary foreground image after background subtraction for (a).

(c) (d)

Figure 3.3(c) Typical scene to be processed (d) Background Image

Finally, the foreground image of human motion is estimated after background subtraction and morphological filtering is obtained and shown in Figure 3.4(a) and Figure 3.4(b)



(a) (b)

Figure 3.4 (a) Foreground Image after background subtraction (b) Foreground image after Morphological Filtering

Figure 3.5 and Figure 3.6 shows the histogram distribution of RGB and H (Hue) and S (Saturation) channels.

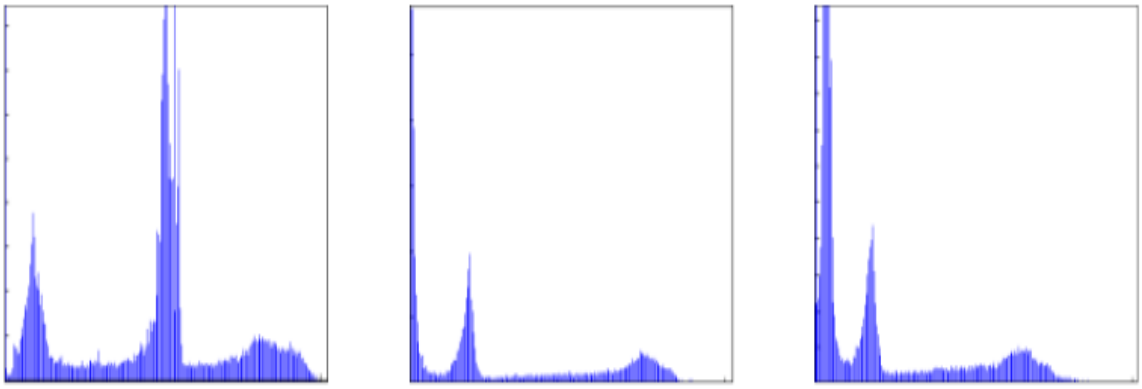


Figure 3.5 Histogram distributions for R, G and B channel

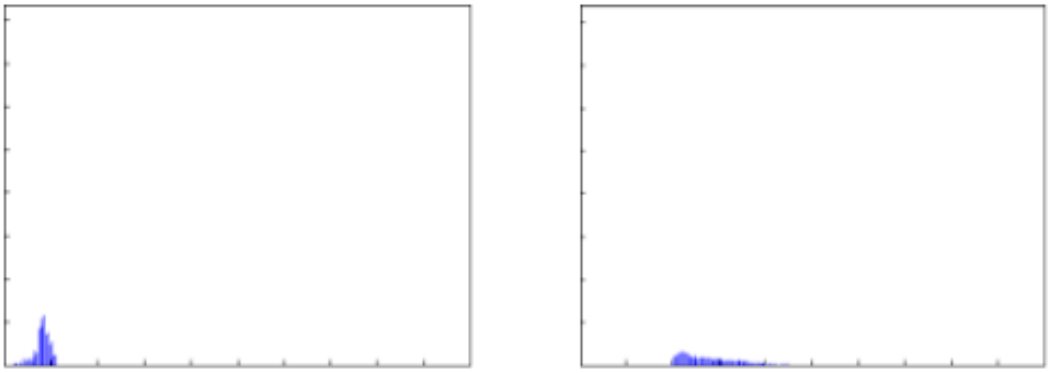


Figure 3.6 Histogram distributions for H and S channel

3.4 ARM PROCESSOR ARCHITECTURE

Motion tracking systems needs to model the mapping from observation space to state space due to the high dimensional characteristics. The standard ARM processors have efficient computing performance to suit high frequency applications. This approach is extensively more compact to solve a real-time problem. An intensive computation is required to process non-viable sampled sequences in real-time. The ARM processor is mainly suitable for embedding a large process with better computing performance.

The real-time computing processor is identified and designed as a reconfigurable hardware with more scalability. The basic computational stages can represent the different steps of the proposed system and process human motion applications. The user interface, hardware controller for memory, VGA visualization and input camera interface have been embedded on the STM32 ARM processor. The system contains a high frame-rate camera as an input device and the different elements that form the system are represented in Figure 3.7. The thin dotted line expresses the data flow processing and those stages are summarized as follows:

a) S_0 - Pre-processing stage.

b) S_1 - The temporal filter computes the temporal derivative and space-time smoothed images.

c) S_2 - Spatial derivatives and complementary pre-processing operations.

d) S_3 - Construction of optimal matrices for integration of neighborhood velocities estimations.

e) S_4- Custom floating-point unit. Human motion estimation requires the computation of a matrix inversion, which includes a division operation. At this stage the resolution of the incoming data bits is significant and expensive arithmetic operations are required. Thus fixed-point arithmetic becomes unaffordable, prompting to design a customized floating-point unit.

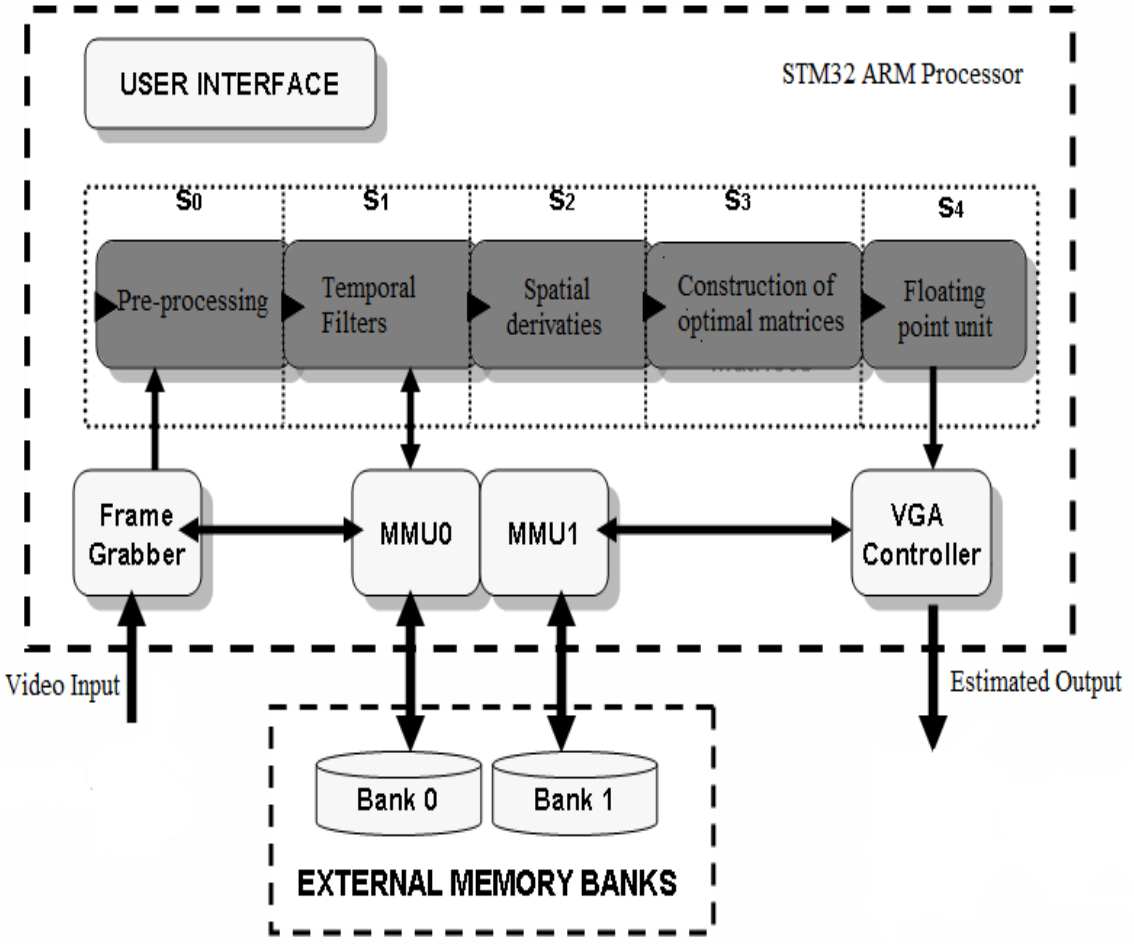


Figure 3.7 Structure of STM32 ARM Processor

3.4.1 Critical Design Factors

The following are the critical factors identified in stage S_4 (Floating Point Unit):

- System Frequency.
- System resources.
- Accuracy.

3.4.2 Steps for ARM Processor

The standard STM32 ARM processor classified into four steps to generate the output is as follows: (refer Figure 3.8)

Step 1: Self generation of input files.

Step 2: After identification of 'C' files and .ASM files in step 1, then generate Hex files.

Step 3: USB interface between the hardware and microvision computer.

Step 4: The hardware will execute with input from the sensors and perform real-time human motion tracking.

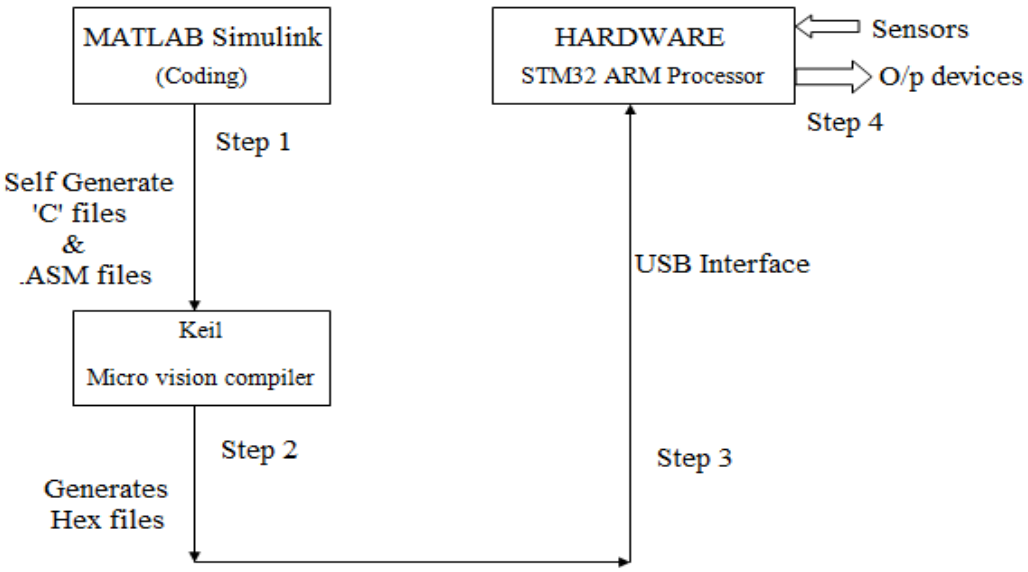


Figure 3.8 Steps for ARM processor generation

3.5 MODULE FOR HUMAN MOTION ESTIMATION

The module for human motion estimation experiment is taken from STM32 family. The STM32 family of 32-bit Flash microcontrollers based on the ARM Cortex™-M processor is designed to offer new degrees of freedom to MCU users. It offers 32-bit product range that combines high performance, real-time capabilities, digital signal processing, and low power, low voltage operation, while maintaining full integration and ease of development. The unparalleled and large range of STM32 devices, based on an industry-standard core and accompanied by a vast choice of tools and software, makes this family of products the ideal choice, both for small projects and for entire platform decisions.

The STM32F103xx (Figure 3.9(a) and Figure 3.9(b) medium-density performance line family incorporates the high-performance ARM Cortex™-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (Flash memory up to 128 Kbytes and SRAM up to 20 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, three general purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: up to two I^2Cs and SPIs, three USARTs, an USB and a CAN. The devices operate from a 2.0 to 3.6 V power supply. They are available in both the –40 to +85 °C temperature range and the –40 to +105 °C extended temperature range. A comprehensive set of power-saving mode allows the design to suit low-power applications.

The STM32F103xx medium-density performance line family includes devices in six different package types: from 36 pins to 100 pins. Depending on the device chosen, different sets of peripherals are included. These features make the STM32F103xx medium-density performance line microcontroller family suitable for a wide range of applications such as motor drives, application control, medical and handheld equipment, PC and gaming peripherals, GPS

platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms and HVACs.

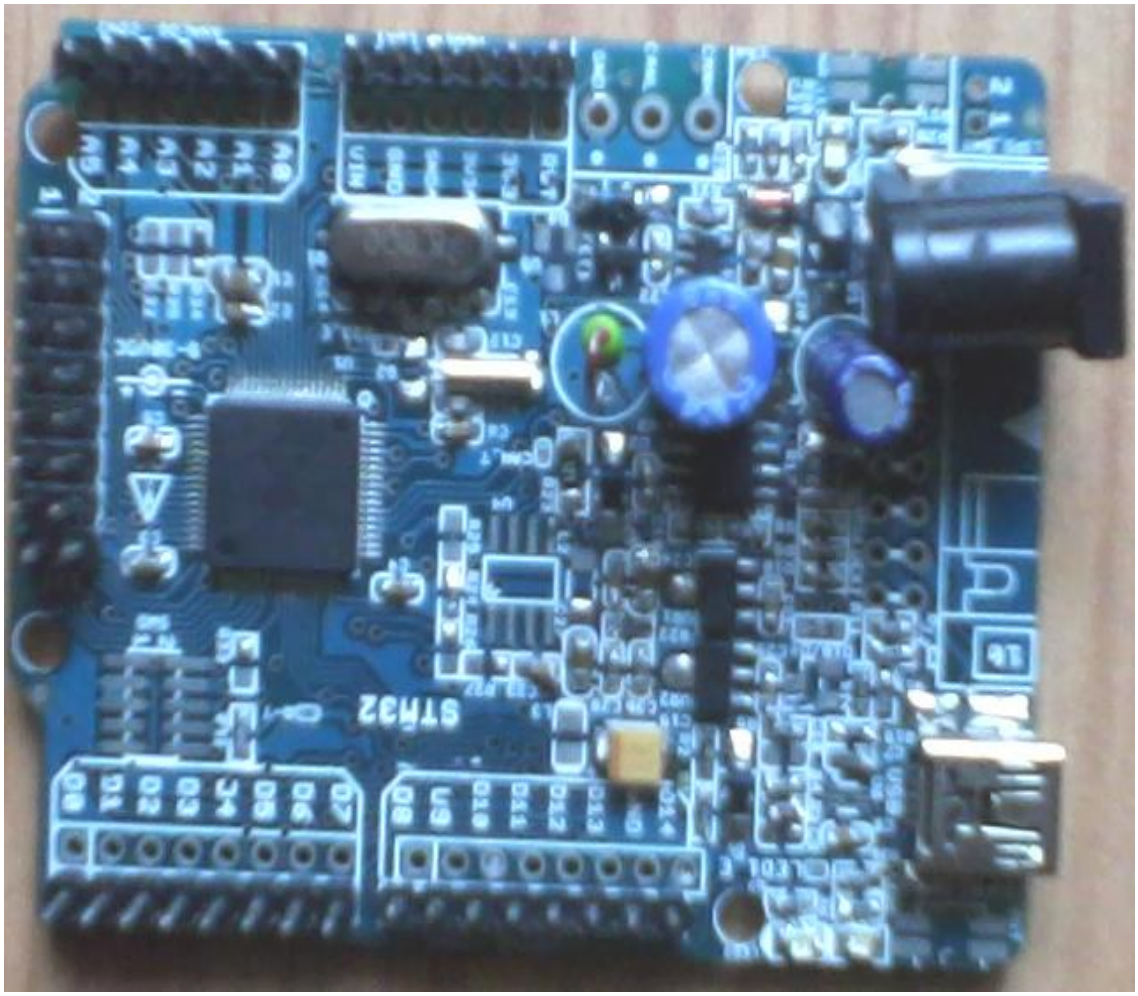


Figure 3.9 (a) Circuit module of STM32F103RB

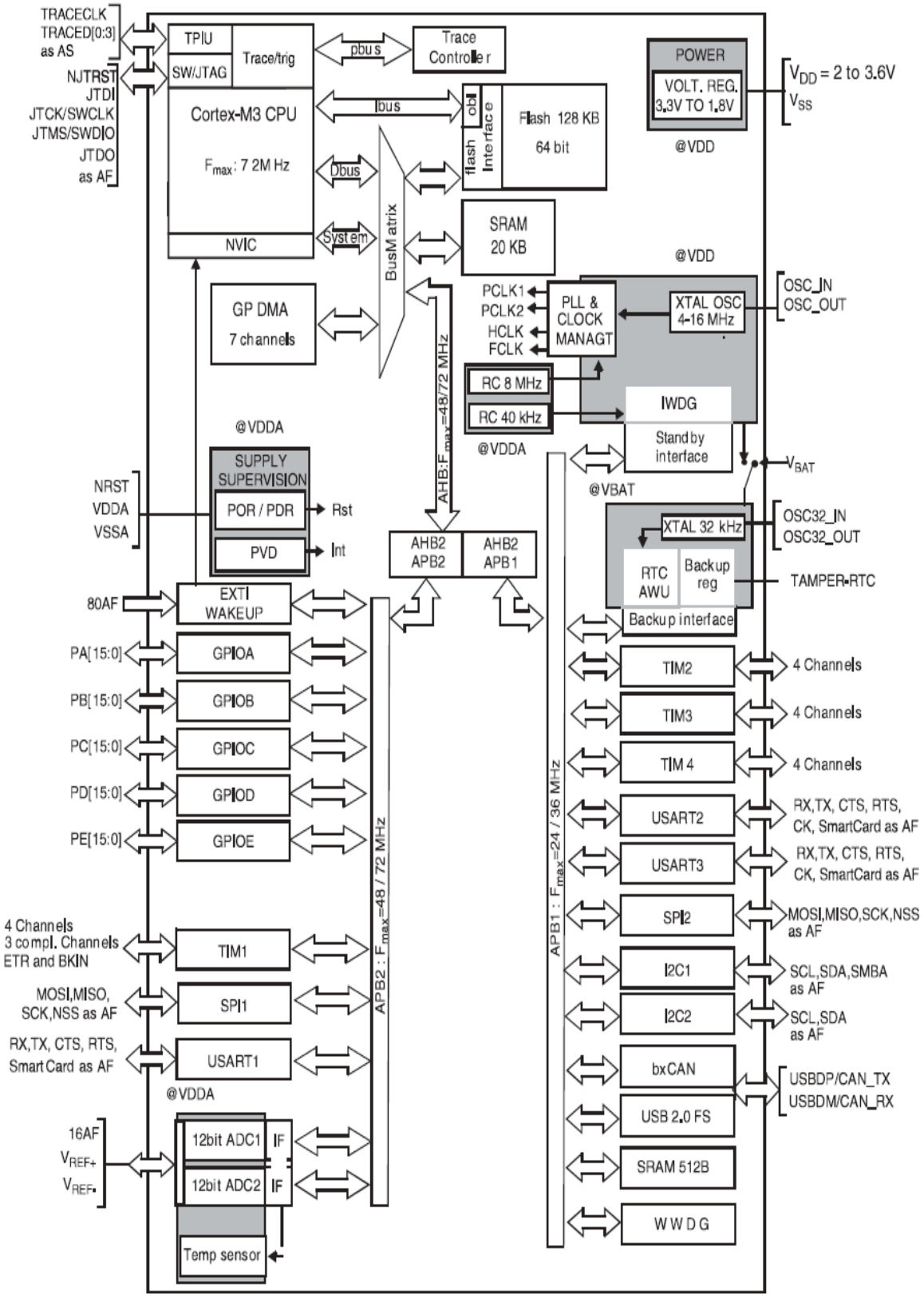


Figure 3.9 (b) Block Diagram for STM32F103RB

3.5.1 Key Features of STM32F103RB

- **ARM 32-bit Cortex™-M3 CPU Core**

  72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performances at zero wait state memory access.
  Single-cycle multiplication and hardware division

- **Memories**

  64 or 128 Kbytes of Flash memory
  20 Kbytes of SRAM

- **Clock, reset and supply management**

  2.0 to 3.6 V application supply and I/Os
  POR, PDR, and programmable voltage detector (PVD)
  4-to-16 MHz crystal oscillator
  Internal 8 MHz factory-trimmed RC
  Internal 40 kHz RC
  PLL for CPU clock
  32 kHz oscillator for RTC with calibration

- **Low power**

  Sleep, Stop and Standby modes
  VBAT supply for RTC and backup registers

- **2 x 12-bit, 1 µs A/D converters (up to 16 channels)**

  Conversion range: 0 to 3.6 V
  Dual-sample and hold capability
  Temperature sensor

- **DMA**

  7-channel DMA controller
  Peripherals supported: timers, ADC, SPIs, I^2Cs and USARTs

- **Up to 80 fast I/O ports**

  26/37/51/80 I/Os, all mappable on 16 external interrupt vectors and almost all 5 V-tolerant

- **Debug mode**

  Serial wire debug (SWD) & JTAG interfaces

- **7 timers**

  Three 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input.

  16-bit motor control PWM timer with dead-time generation and emergency stop

  2 watchdog timers (Independent and Window)

  SysTick timer 24-bit downcounter

- **Up to 9 communication interfaces**

  Up to 2 x I^2C interfaces (SMBus/PMBus)

  Up to 3 USARTs (ISO 7816 interface, LIN, IrDA capability, modem control)

  Up to 2 SPIs (18 Mbit/s)

  CAN interface (2.0B Active)

  USB 2.0 full-speed interface

- **CRC calculation unit, 96-bit unique ID**

- **Packages are ECOPACK®**