

Research and Implementation of an STM32-based Shuttle-run test system for athletes

Jones Sophie^{1, a}, Roberts George^{1, b}, Thomas Patricia^{2, c, *},

¹University of Exeter, Stocker Road, Exeter, Devon, EX4 4PY, UK

²Lloyd's Building, 1 Lime Street, London, EC3M 7HA, United Kingdom

a.jones7_sophie_2026@icloud.com, b.george_5110214@163.com, c.thomaspatricia_09_@yahoo.com

*Corresponding Author

Abstract: Starting from the main functional requirements of the test system, this paper expounds the design concepts of both software and hardware for an electronic shuttle-run test system aimed at athletes, along with the key technical principles and implementation methods involved. The system employs an STM32 microprocessor for overall control, accurately measures an athlete's shuttle-run performance, and presents the results prominently on a running recorder. It not only eliminates human errors inherent in manual timing but also transmits the data wirelessly to a PC host for further statistical analysis.

Keywords: STM32, automatic measurement, Wi-Fi, sensor

1. Introduction

With the continuous development of basketball, electronic test systems have entered the daily training and comprehensive evaluation routines of players. The shuttle run—one of the core drills—requires two markers placed 15 m apart; athletes sprint from one marker to the other, turn, and repeat. Conventional hand-held stopwatch timing introduces large errors: the official cannot precisely judge whether the athlete reaches the marker, and human reaction time adds further uncertainty [1,2].

The electronic shuttle-run test system developed by the authors employs photoelectric sensors to record both the number of shuttles completed within a set period and the exact time taken. Results are displayed prominently on a portable running recorder. An on-board Wi-Fi module transmits all data to a PC host for storage, statistical analysis, and long-term tracking.

2. Overall system architecture

The test set-up consists of a running recorder, two running-sensor measurement posts, two post chargers, a Wi-Fi module, and a PC host. The system block diagram is shown in Fig. 1.

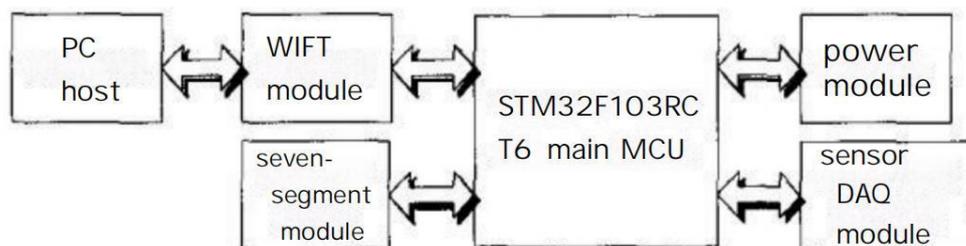


Figure 1 System Block Diagram

The microcontroller selected is the STM32F103RCT6, a mainstream ARM device from STMicroelectronics built around the ARM Cortex-M3 core. Compared with traditional 8051 families, it offers richer resources, more powerful peripherals, and higher processing bandwidth while maintaining high speed and low power. The chip integrates 256 KB of Flash and 48 KB of SRAM, and provides a rich feature set—DMA, motor-control PWM, PDR/POR/PVD, on-chip temperature sensor, independent watchdog, SWD debug support, etc.—greatly reducing external circuitry [3,4].

The test logic relies on two photoelectric sensors (upper and lower) inside each running-sensor post to detect turn-around events; results are shown on the running recorder. The recorder carries three rows of three-digit common-cathode eight-digit LED displays (lap count, seconds, and 0.01 s). Display driving is handled by the ZLG7289, which provides an SPI-compatible serial interface and can drive up to eight common-cathode digits simultaneously. After a test, the recorder transmits the data to the PC host via Wi-Fi. The system employs the industrial-grade USR-WIFI232-A module with an on-board antenna. It performs transparent serial-to-Wi-Fi conversion: the MCU treats it as an ordinary UART, while the PC sees it as a standard TCP/IP device, so normal socket programming is sufficient for communication.

All sensing is non-contact; therefore sensor reliability and detection range are critical. The system uses the GC16-M2000-D1 diffuse-reflective photoelectric switch from Shanghai Run-An Industrial Sensor Co., offering a maximum sensing distance of 2 m.

3. Hardware design

3.1. Minimum system for the STM32 MCU

The minimum system comprises power, reset, debug/download interface, and crystal oscillator. The core runs at 3.3 V, but a 7.2 V Li-battery (VBAT) is used in the recorder; hence a buck regulator is included. An 8 MHz crystal provides the main clock.

Because the STM32 is reset on a low level, the external reset circuit pulls the RESET pin to GND during reset and to 3.3 V during normal operation; capacitor C37 filters noise. The debug interface follows the standard JTAG footprint and also supports SWD mode. The schematic is given in Figure 2 [5,6].

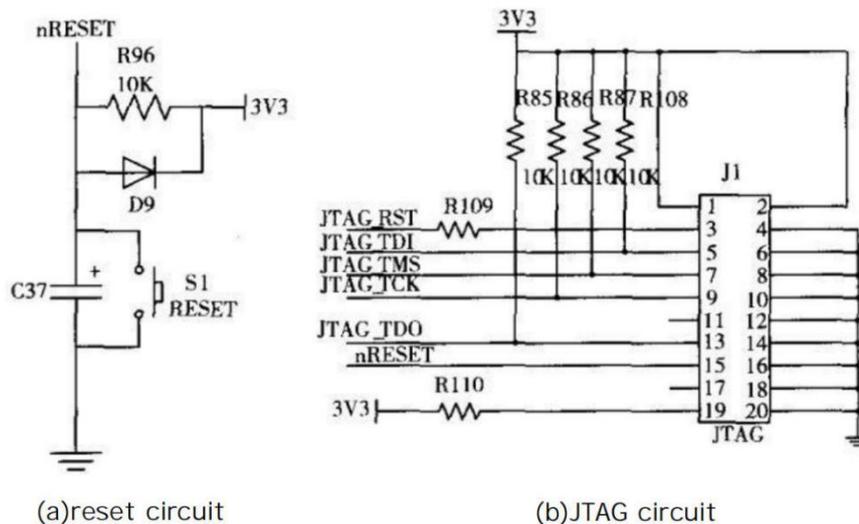


Figure 2 Reset and JTAG Circuits

3.2. Sensor data-acquisition module

The system employs the GC16-M2000-DN1 photoelectric sensor from Shanghai Run-An Industrial Sensor Co. Key specifications: supply 10–30 VDC, load current ≤ 200 mA, maximum sensing range 2 m. It

is a DC three-wire, normally-open, PNP-output device: in the absence of a target the output line floats (open); when a target is detected the output is pulled to VCC (high).

The sensor is used in “reflector mode”: a retro-reflective foil is mounted opposite the sensor. When an athlete passes between the sensor and the reflector the returned beam is interrupted, generating a high-level pulse that marks the turn-around event [7,8]. The detection principle is illustrated in Figure 3.

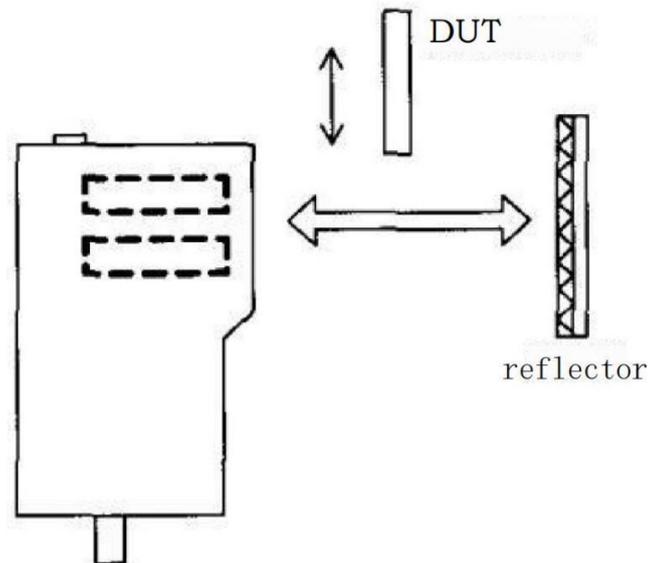


Figure 3 Retro-reflective Detection Scheme

In the system two sensor posts are positioned 15 m apart. Inside each post two photoelectric switches are mounted one above the other and wired in parallel. Because athletes differ in running style—some reach the turn-around line with their toes first, others with their shins—the dual arrangement guarantees that any part of the body breaking either beam will be detected. When either sensor is obscured its output is coupled to the MCU through an opto-isolator to increment the lap counter; hence the two switches are paralleled. Figure 4 shows the mechanical layout of a running-sensor post.

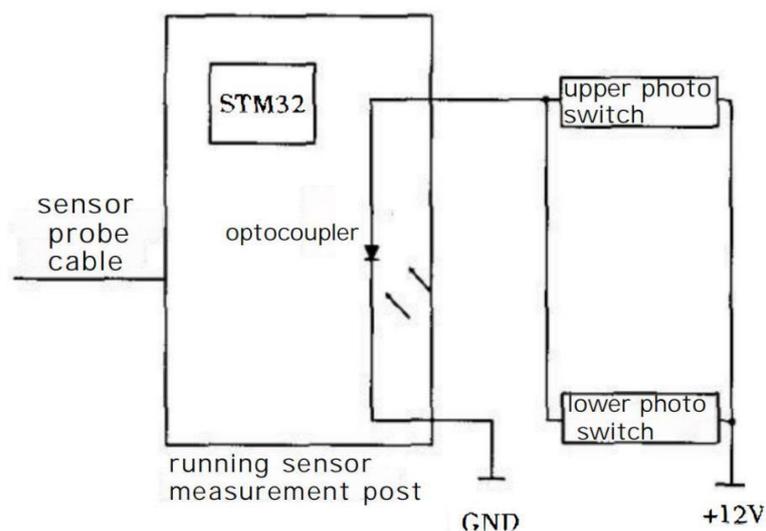


Figure 4 Running-Sensor Measurement Post Diagram

3.3. Seven-segment display module

To present the athlete’s performance in real time, the system shows lap count, seconds, and 0.01-s resolution on a bright, highly visible panel. Because the data are purely numeric, eight-digit common-cathode seven-segment LEDs are used (Figure 5). Each digit consists of eight LEDs (seven segments plus decimal point); applying current to the appropriate pins illuminates the required segments to form the numbers.

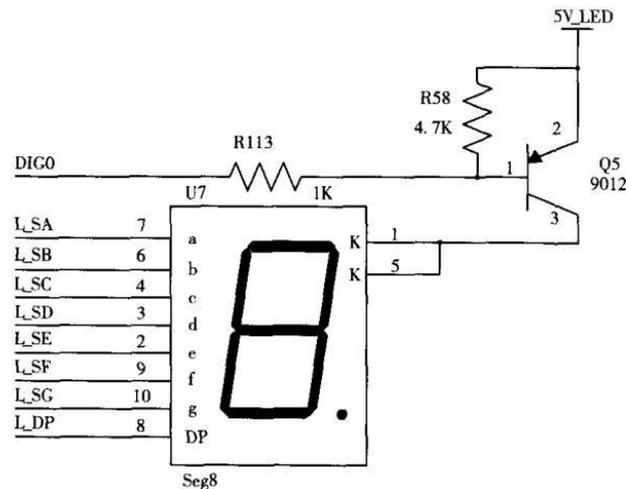


Figure 5 Seven-Segment Display Circuit

3.4. Seven-segment display module (continued)

Eight-digit displays can be driven either statically (DC) or dynamically (multiplexed). This design uses dynamic driving: the common-cathode lines (COM) of all digits are enabled sequentially in time slots, so only one digit is on at any instant, but the cycle is fast enough for flicker-free viewing. With limited MCU I/O, multiplexing controls more digits while saving pins and simplifying external circuitry [9].

The driver is ZLG7289A, a serial-interface chip that supports up to eight common-cathode digits and offers commands for blanking, blinking, left/right shifting, and individual segment addressing. ZLG7289A requires an external 12 MHz crystal. Because three data fields need nine digits total, two chips are used: one drives six digits, the other drives three. The ZLG7289A connection is shown in Figure 6.

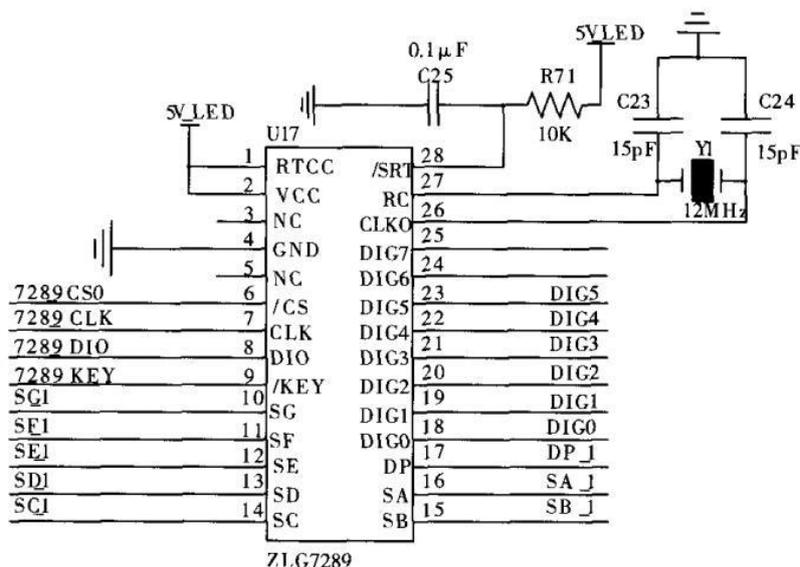


Figure 6 ZLG7289A Circuit Diagram

3.5. Wi-Fi wireless communication module

After each trial the system must send the result to the PC host for storage and analysis. Wi-Fi was chosen for three reasons: Radio coverage is far greater than Bluetooth. Data rate (up to 11 Mbps) guarantees real-time delivery. Low module price keeps total cost down. The Wi-Fi module interface is shown in Figure 7.

4. Software design

The firmware is written in C and developed under Keil μ Vision 4. Main routines cover system initialization, display, timing, sensor-signal reception, and PC communication. The flow chart is given in Figure 8 [10]. Power-on sequence: Initialize MCU (clock, GPIO, timers, UART, Wi-Fi). Self-test sensors and display. Wait for athlete.

When the athlete first cuts the beam at Post-1, timing starts and the display updates. Posts 1 and 2 then alternate—each new beam break increments the lap counter; the same post cannot register two consecutive breaks. When lap count reaches 17, timing stops and the MCU transmits the result to the PC host for logging.

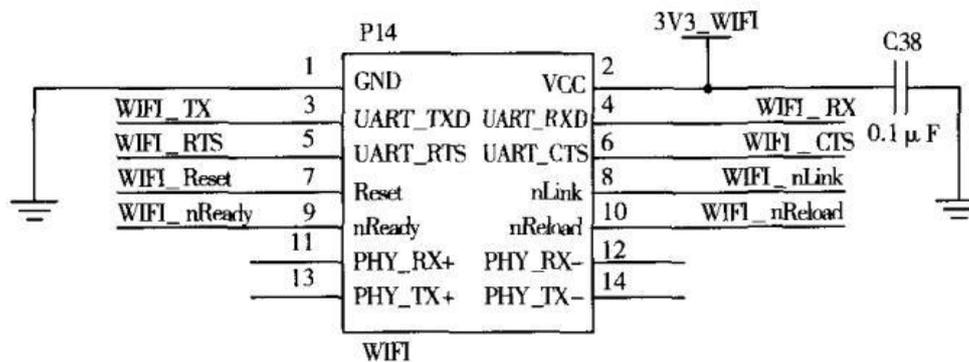


Figure 7 Wi-Fi Module Circuit Diagram

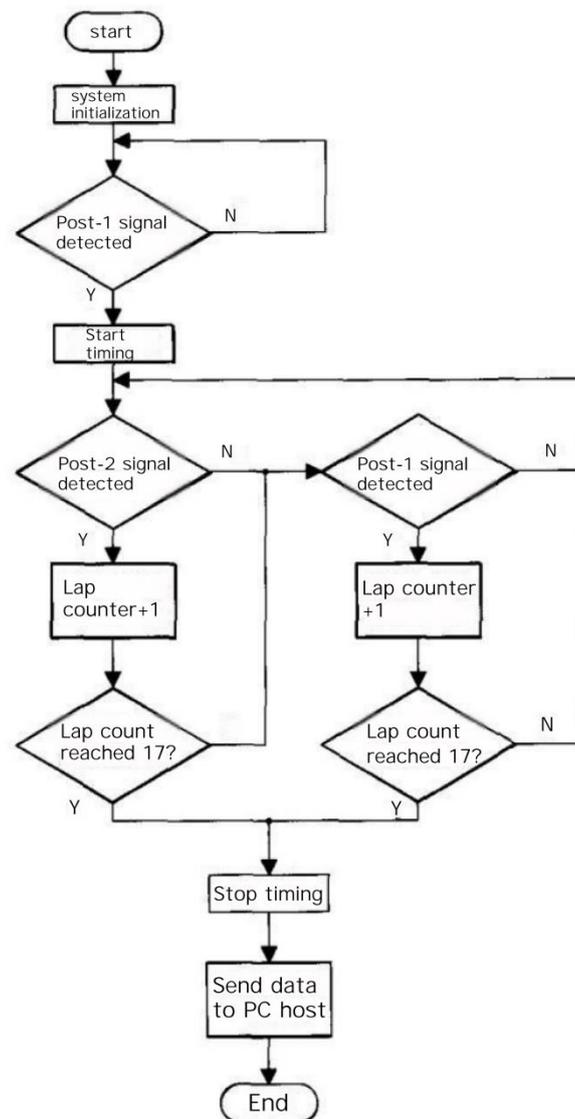


Figure 8 Software Flowchart

5. Conclusion

Centered on the STM32F-series ARM microcontroller, the system integrates power management, wireless communication, LED display, and sensor detection to provide fully automatic timing for basketball shuttle-run tests. It delivers automatic start/stop, high-visibility results, and wireless data upload. The developed unit improves fairness, accelerates throughput, and significantly reduces manual effort in athlete evaluation.

6. References

- [1] Lian Bizhen, Song Wei, Shan Shuguang. Research on the application of basketball skill and physical-fitness evaluation systems [J]. *Journal of Beijing Sport University*, 2013, 36(6): 115-118.
- [2] Lu Hongwu, Tan Jinbo. Development of an automatic test system for basketball-specific examinations [J]. *Journal of Shandong Normal University (Natural Science)*, 2013, 18(1): 36-38.
- [3] Liu Jun. *STM32 Case Studies* [M]. Beijing: Beihang University Press, 2011: 11-25.

- [4] Wang Yonghong. Principles and Practice of STM32 ARM Cortex-M3 Microcontrollers [M]. Beijing: Beihang University Press, 2008: 14-82.
- [5] Joseph Yiu. The Definitive Guide to ARM Cortex-M3 [M]. Translated by Song Yan. Beijing: Beihang University Press, 2009: 100-106.
- [6] Yang Guangxiang, Liang Hua, Zhu Jun. STM32 Microcontroller Principles and Engineering Practice [M]. Wuhan: Wuhan University of Technology Press, 2013: 18-28.
- [7] Kang Yihua. Engineering Measurement Technology [M]. Beijing: China Machine Press, 2005: 73-82.
- [8] Yang Bangwen. Modern Sensor Applications Handbook [M]. Beijing: China Machine Press, 2006: 378-397.
- [9] Li Qunfang, Zhang Shijun, Huang Jian. Single-Chip Microcomputers and Interface Technology [M]. Beijing: Publishing House of Electronics Industry, 2008: 97-100.
- [10] Li Ning. Getting Started with ARM Development Tools Real View MDK [M]. Beijing: Beihang University Press, 2008: 2-7.