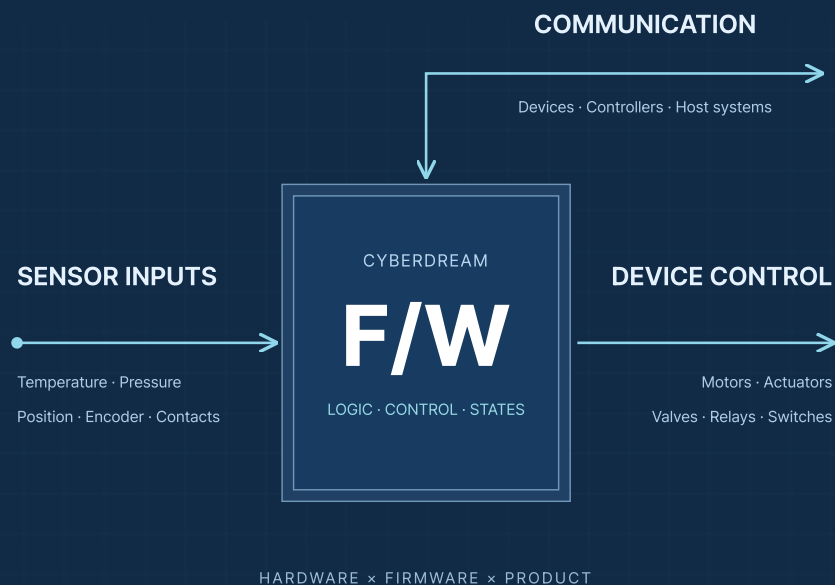


Embedded Firmware

From hardware potential to product behavior.

Interpret sensor inputs and control external devices.

We develop and improve MCU firmware based on your circuits, boards and existing code. We connect sensor processing, device control and wired or wireless communication, and design state transitions, fault handling, shutdown and recovery behavior.



CONTROL APPLICATIONS

Turn input signals and commands into product behavior.

Control devices through their drive circuits and interfaces.

01 ANALYSIS & DESIGN

Review requirements, circuits and existing code.

Define I/O, state transitions and fault handling.

02 DEVELOPMENT & VERIFICATION

Develop and improve control and communication.

Verify hardware–firmware integration and behavior.

03 REPRODUCIBILITY & HANDOVER

Deliver build setup, source code and documentation.

Provide agreed test results and outstanding issues.

01 / ENGINEERING DEPTH

Six disciplines behind device behavior

Beyond individual drivers, we define control timing, data flow, exception handling and the way people operate the product.

01

MOTION CONTROL

Motor & encoder control

Review PWM control, encoder feedback and PID timing. Define acceleration, deceleration, direction changes, homing and missing-pulse detection for the device.

PWM · Timer · Encoder · PID

02

SENSING & ACQUISITION

Sensors & ADC acquisition

Review sampling, channel switching, filtering, contact detection and calibration. Align firmware acquisition timing with the circuit's signal-settling requirements.

ADC · MUX · Filtering · Calibration

03

DEVICE CONNECTIVITY

BLE & device communication

Define command/status packets, timing, acknowledgements, timeouts and retries. Address disconnection behavior and state consistency after reconnection.

BLE · UART · SPI · I2C

04

PRODUCT BEHAVIOR

State machines & device UI

Separate start, pause, completion and fault states. Align short/long button presses, displays, alerts and battery indicators with actual device behavior.

State machine · Button · Display · Buzzer

05

FAULT HANDLING

Output control & exceptions

Define sensor-fault, low-voltage and charging conditions, plus watchdog/reset recovery. Consolidate output control and review conflicting commands.

Interlock · Watchdog · Reset · Power state

06

CODE MODERNIZATION

Legacy code & refactoring

Identify active build sources and execution paths. Separate BSP, drivers and product logic; document board revisions, pin assignments, build settings and changes.

C/C++ · BSP/HAL · Build · Regression

Start with your MCU and development environment

Our discussions draw on nRF52/AT32 review and development materials and STM32/ESP32 sensor-board review work. MCU, SDK, toolchain and peripheral support are confirmed before work begins.

02 / APPLICATIONS

Products and engineering challenges we can scope

These service examples generalize technical challenges addressed in our review materials. Implementation and testing scope is agreed for each product.

01 / CONNECTED MOTION

Wireless motor-driven devices

Products that must align target speed, actual motion and status across a controller and wireless device.

- Encoder scaling, loop timing, speed ramps and direction changes
- BLE commands, responses, disconnects and reconnection
- Consistent controls, display, battery and motion states

What we document

Control sequences · protocol definitions · verification items

02 / MULTI-MODE DEVICE

Equipment with multiple operating modes

Products that need coordinated user commands, sensor conditions and output sequences.

- Mode selection, start, pause and shutdown states
- Output permission under contact, charging and sensor conditions
- Separate UI and output management; define fault recovery

What we document

State transitions · output conditions · requirements-to-test matrix

03 / SENSOR ACQUISITION

Multi-channel sensor boards

Systems whose analog input characteristics and MCU acquisition settings must work together.

- ADC sampling, MUX switching and signal-settling review
- I2C sensor and USB interface requirements against the circuit
- Startup, power sequencing, acquisition and transfer requirements

What we document

H/W-F/W interface conditions · open questions · test plan

03 / HARDWARE-FIRMWARE REVIEW

Find the gaps between circuits and code

Matching pins do not guarantee matching behavior. We compare schematics, component documentation, active sources and build artifacts to identify integration issues.

01

Pin mapping & control meaning

What does a stop command actually switch off?

Compare GPIO names, polarity, driver truth tables and power paths. Distinguish output-level changes, motor stops and actual power disconnection.

02

Active build & execution paths

Does the function under review actually run?

Check project settings, linker maps and call paths to identify active code. Review whether legacy files and comments describe the implementation in use.

03

Sensing, timing & units

Do measurements and control timing agree?

Compare encoder scaling, gear ratio, PID timing and ADC conversion. Review the combined delay from circuit filtering, sampling and accumulated fault decisions.

04

Fault stops & recovery

What state follows a disconnect or reset?

Review missing pulses, communication loss, charging, watchdog and reset paths. Separate software stopping from hardware cutoff and define restart conditions.

REVIEW → PRIORITIZE → RETEST

Turn findings into an actionable correction plan

Deliver circuit-to-code cross-references, evidence locations, correction priorities and retest criteria. Identify open items with the BOM, revision, traces or logs needed to resolve them.

Document/code findings are distinguished from physical measurements. Board verification is separately performed with the agreed hardware, equipment and test conditions.

04 / ARCHITECTURE & VERIFICATION

Separate product decisions from hardware control

Clear responsibilities make product behavior easier to trace and test when the board changes.

01 Product behavior

Modes · state transitions · user actions

02 Control & communication

Algorithms · packets · output permission

03 BSP & drivers

Pin mapping · timers · ADC · peripherals

04 Board & devices

MCU · motor · sensors · display

BEHAVIOR DESIGN

Design the exception paths, too

01 / Ready

02 / Active

03 / Stop & check

04 / User resumes

Example: stop output on a sensor fault or disconnection, then define recovery conditions and an explicit resume action. The policy is agreed for each product.

Evidence with a clear verification scope

01**Code & circuit review**

Requirements, pin assignments, active sources and execution paths

02**Reproducible build**

Toolchain, SDK, settings, binaries and change history

03**Host & regression tests**

State transitions, boundaries and exceptional inputs

04**Board & integration tests**

Measurement conditions, instruments, traces and acceptance criteria

Activities depend on the agreed scope and available test setup. Code, build and host-test results are reported separately from physical-board measurements.

05 / PROCESS & HANDOVER

Deliverables your next development cycle can use

01

Review inputs

Check requirements, schematics, board revisions, source code and build setup.

02

Define behavior

Agree on states, control, interfaces, exceptions and acceptance criteria.

03

Develop & test

Implement and run regression tests, then verify on boards under agreed conditions.

04

Hand over

Deliver source, binaries, documentation, test results and outstanding issues.

01 Source & structure

Agreed source code, module responsibilities and changes

02 Build environment

Toolchain/SDK versions, build options and instructions

03 Binaries & programming

Board-specific releases and programming instructions

04 Behavior & interfaces

States, pin assignments, protocols and output conditions

05 Test evidence

Agreed test conditions, results, traces and logs

06 Open issues & handover

Unverified items, reproduction steps and follow-up actions

DEFINED IN YOUR PROPOSAL

Agreed deliverables

The proposal specifies source code, build configuration, binaries, behavior/interface documents, changes and test results. SDK and third-party sources follow their applicable terms.

06 / SCOPE & CONTACT

Your board. Your next product milestone.

We will help define a starting point: new firmware, an existing codebase or a focused improvement.

Quoted after review

Pricing depends on source/build availability, MCU and board count, control and interface complexity, code condition and physical test requirements.

Separately scoped work

Circuit redesign, PCB fabrication, prototypes, test fixtures, accredited testing, certification and on-site support are scoped and priced separately.

Clearly defined verification

Code reviews, builds, host tests and physical-board tests are reported separately, with unverified items and further checks identified.

Start with the materials you have

- + Product purpose, desired behavior or observed symptoms
- + Schematics, MCU, board revision and relevant datasheets
- + Existing source, SDK, build instructions and protocol specifications
- + Target schedule, deliverables and board/test equipment availability

LET'S BUILD TOGETHER

Discuss firmware development

swhong@cyberdream.biz

+82 32 832 1361 · cyberdream.co.kr

Discuss firmware development



Online inquiry: Embedded firmware