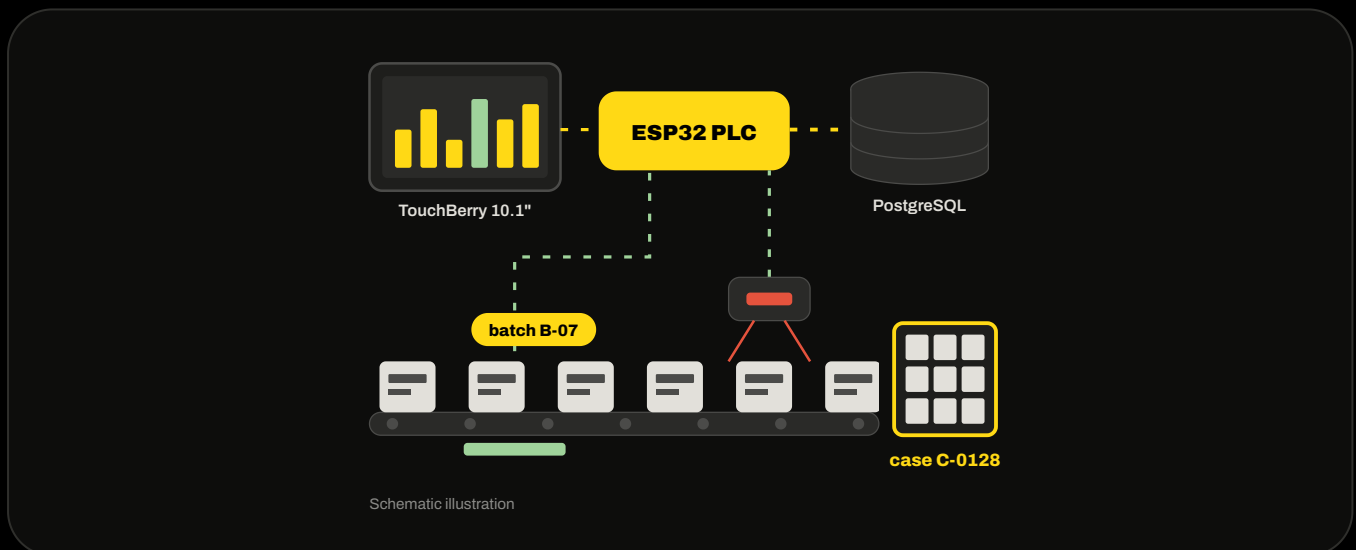


FOOD PACKAGING · ESP32 PLC

Every pack, every batch, every case. Traced by ESP32 PLC.

A food packaging company needed automatic traceability on three lines that ran at full capacity. Weight sensors, a barcode scanner and ESP32 PLCs now record units, weight, batch and case for every product, and send the data to a central PostgreSQL database.

[Read the interactive case study online ›](#)

3

packaging lines digitised

6

data types recorded automatically

2

packaging levels traced: pack and case

PostgreSQL

central database for every record

THE CHALLENGE

Trace every pack to its batch and its case.

The three lines already ran at full capacity. What they could not do was collect their data in an automated way. Part of it was written by hand, and someone typed it into the ERP later. The end customer had one essential requirement: automated traceability of every product.

01

Data by hand

Full capacity, but part of the data was collected by hand. That costs competitiveness and opens the door to human error.

02

Two packaging levels

Each primary pack has a barcode and a batch number. Each case groups a required number of packs and needs its own number.

03

Many formats

Multiple products in various packaging formats. The data must be up to date, without waiting for someone to enter it.

04

Late data for the ERP

Before, the data was not available in real time. Manual entry into the ERP could add errors.



Automated traceability was not optional

It was the one essential requirement of the end customer: identify every product and every grouping, and save the production process, with no manual step.

HOW IT WORKS

Three loops: count, group, move.

Loop 1 · Count and weigh

Weight sensors count the units and the packed weight.

Weight sensors on the line tell the PLC two things: how many units passed and how much product was packed. Each product also carries the batch number that the marking system sends to the PLC. The database uses these records to calculate the quantity packed in each batch. Nobody counts by hand.

- Weight sensors: total units and total packed weight
- Marking system: a unique batch number for each packed product, sent to the PLC
- The database calculates the packs in each batch

Loop 2 · Batch to case

A case gets its number only when it is complete.

A counting system counts the packs that go into each case and compares the count with the required quantity per case. When the case is complete, it gets a unique case number. The PLC records it, together with the batch, and sends it to the central database. A barcode scanner connected to the PLC reads the barcode of each grouping.

- Count per case against the required quantity
- Unique case number, recorded in the PLC and the central database
- Barcode scan of each grouping in a case

Loop 3 · Move the belt

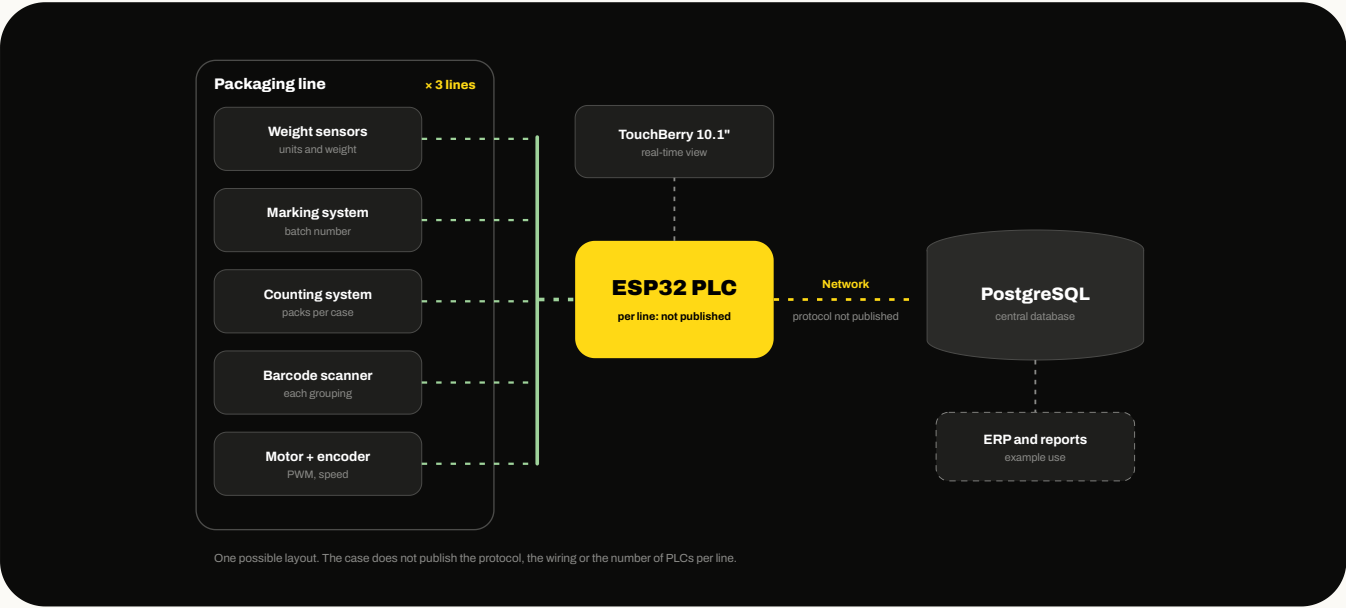
PWM drives the conveyor. An encoder confirms the speed.

The conveyor belts are controlled by PWM and encoders. A typical way to do it: the PLC sets a PWM duty for the motor, reads the encoder pulses and corrects the duty until the belt runs at the set point. In the panel, change the format or add load to the belt, and the loop corrects itself.

- PWM output to the conveyor motor (example duty values)
- Encoder input to measure the belt speed
- A set point per packaging format (example)

How does an ESP32 PLC trace a packaging line?

On each of the three lines, ESP32 PLCs read the weight sensors, the batch number from the marking system, the case count and the barcode scanner. They control the conveyor with PWM and an encoder, and they send every record to a central PostgreSQL database. A 10.1" TouchBerry touch screen shows the production status in real time. The case does not publish the protocol, the wiring or how many PLCs each line uses.



Records to the database Signals from and to the line Example elements, not published by the case

Signals and data

Data	Between	Type	Purpose
Units and weight	Weight sensors → ESP32 PLC	Sensor	Counts the units and the total packed weight
Batch number	Marking system → ESP32 PLC	Data	A unique batch number for each packed product
Case count	Counting system → ESP32 PLC	Count	Compares the packs in a case with the required quantity
Case barcode	Barcode scanner → ESP32 PLC	Reader	Reads the barcode of each grouping in a case
Belt speed	ESP32 PLC ↔ motor and encoder	PWM + encoder	Controls the conveyor belts
Records	ESP32 PLC → PostgreSQL	Network	Units, weight, batches, cases and barcodes. Protocol not published
Live view	TouchBerry 10.1" touch screen	Display	Shows the production status in real time

ESP32 PLC to trace. TouchBerry to see.



ESP32 PLC

Industrial controllers built on the dual core ESP32, with Ethernet, Wi-Fi and RS-485, for your DIN rail. On this line they read the sensors, the marking system and the scanner, drive the conveyor with PWM and send every record to the database.

Role	Counting, batch and case records, conveyor control
Processor	ESP32 dual core
Network	Ethernet 10/100, Wi-Fi 2.4 GHz, Bluetooth LE
Serial	RS-485, RS-232
Programming	C++ with the Arduino IDE
Family	ESP32 PLC 14, 19R, 21, 38AR, 38R, 42, 57R and 58

The case does not publish the exact ESP32 PLC model or how many units each line uses. The installation scheme shows two ESP32 PLC units. The image shows the ESP32 PLC 21 FLEX.

[See the ESP32 PLC family >](#)



TouchBerry 10.1" touch screen

A 10.1" TouchBerry touch screen appears in the installation scheme, next to the ESP32 PLC. It gives the production staff the live view of the line.

Role	Real-time view of the production status
Display	10.1", capacitive touch
Computer	Raspberry Pi, Linux

The case does not publish the TouchBerry version. The image shows the TouchBerry Pi Panel PC 10.1" FLEX.

[See the TouchBerry Pi 10.1" >](#)

Data at the source

The PLC records each batch and case number where it happens, on the line. The database gets the same record a moment later. No retyping.

Ready for any format

The company packs multiple products in various formats with the same system. Example: a set point and a case size per format.

A database that scales

PostgreSQL was chosen because it manages large amounts of data quickly. Other SQL databases, such as SQLite, also work.

Open, with no licence fees

One of the points that won the project. You program the ESP32 PLC in C++ with the Arduino IDE and the Industrial Shields boards package.

Modular

The system can grow later on the same platform: more I/O with expansion modules, and LoRa, LTE-M/NB-IoT or CAN expansion boards, on models with an expansion slot.

Support before, during and after

Our team was available before, during and after the installation, by phone, email or WhatsApp. Equipment designed and manufactured for industrial use.

SIMILAR LINE?

Tell us your line and your codes. An engineer will answer.

Tell us how many lines you have, which codes you read (barcode, QR or RFID), how you build batches and cases, and where the data must go: a database, an ERP or an MES. We will tell you which controller fits and how to connect it.

READ IT ONLINE

industrialshields.com/packaging-line-traceability-case-study

EMAIL

info@industrialshields.com

PHONE

[\(+34\) 938 760 191](tel:+34938760191)