

# POLICE ASSISTANT DEVICE - CLOUD DATABASE FOR CRIMINAL RECORDS

**Dhishanth T D**

Eee Department Panimalar Engineering College  
Chennai ,India

**Dharshan R**

Eee Department Panimalar Engineering College  
Chennai ,India

**Gowtham N**

Eee Department Panimalar Engineering  
College  
Chennai ,India

**Derirck Albino L**

Eee Department Panimalar Engineering  
College  
Chennai ,India

**S.Srinivasan** Associate Professor

Eee Department Panimalar Engineering College  
Chennai ,India

**Abstract**--in iot applications, Database security and safety is one of the important aspects in the area of networks system. The crime related to criminal actions and alert systems has been a tremendous rise in every day. This generates a crucial need for an effective criminal record system. In this project, a compact, efficient system is studied, designed and explored. An integrated UI system, which is to monitors criminal's record and crimes related to them with conditions and the right authentication using fingerprint sensor. In this project, we provide a trustworthy and solid Criminal Record System (CRS) design with elements boosting traffic signal security. In addition to the recognition system, our proposed recording system also includes a number of additional features. Internet of Things technology is one of the key aspects enabled by this system. The system is kept redundant to provide reliability even in the worst- case scenario, however due to budgetary restrictions, a trade-off between redundancy and cost was required. This method is made to work with practically every location that provides crime assistance.

Additional to this we have also implied the concept of powering a vehicle if only the person is not drunk. Alcohol presence is continuously monitored using sensors and whenever alcohol detected, the vehicle engine refuse to start, to avoid accidents.

## I. INTRODUCTION

Police provides safety to citizens. It always remains steady for arresting any criminal who is a threat for the

safety of society. After registering the FIR from any citizen, police starts its work and on that basis it arrests the criminals if proofs are found against them. Once the criminals are arrested, police starts investigation from them. After getting all the proofs against the criminal, it is the duty of the police to present all the proofs honestly to the court so that the right man can get right punishment.

The true and right information provided by the people to police helps a lot in arresting the criminals who try to spoil the peaceful environment of society. Along with low salary scale, facilities of modern technology such as computerized system of keeping records are not provided to police department which causes low efficiency. As it is the age of computers and all the organizations today use computers to maintain their records, so this facility should also be given to police department in order to increase their efficiency and to save their time.

In our Project we are going to implement a CRMS (Criminal Record Management System). This is a database system in which police will keep the record of criminals who have been arrested, to be arrested or escaped. This will help the police department to manage their records easily. In police system when an incident occurs, a petitioner reports an FIR (First Information Report). Police starts investigation according to law on this FIR. An investigation officer supervises the investigation process. The main concerning people in the whole process are petitioner (The person who files an FIR), Victim, Accused/Criminal, investigation officer. 10

II. EXISTING SYSTEM.

The existing system that is being used by police department pertaining to the information of the prisoners, stores the name of the prisoners, information of the crime, date of FIR, background of the criminal and duration of the prison. However, once the period comes to an end, and the individual is released, it becomes difficult for the personnel serving the police forces to keep a track of the date and time of the release of the particular offender. Since the information of the release date of the offender is not present with each and every personnel, they are not notified on time, leading to chaos.

III. PROPOSED SYSTEM

The aim of the proposed system is to cut through the chaos and workload, by innovating the current system. The proposed system notifies each and every personnel with the application on his android phone, about the release of an offender, including the other details. It eliminates the existing issue where the department of police of different area of jurisdiction remains unaware of the release of the criminal and his/her extra details, by creating a notification every time there is a sight of a criminal. This approach can thereby reduce the effort of the police and save their time, which could possibly be used in a productive manner

3.3 BLOCK DIAGRAM

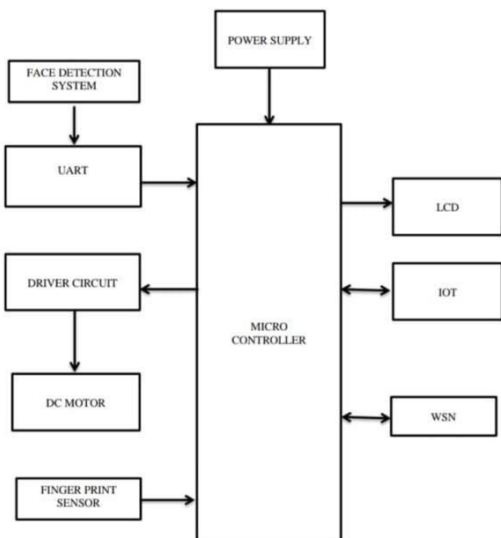


fig:1 Block Diagram of Proposed System

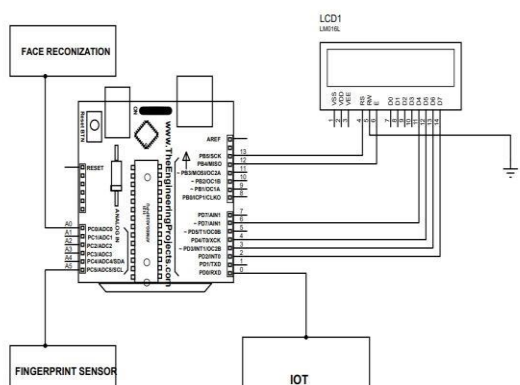


fig :2 Circuit Diagram of Proposed System

Block Diagram Description

- Initially power supply comes from ac to dc adapter (230V ac) converted to 12V dc to the ESP32 Microcontroller.
- IN2596 Buck regulator step-downs 12V dc to 5V for UART and 5V for fingerprint sensor.
- For data transmission and API calling we use the WSN hardware module, so that we use the mobile phone network to connect with the microcontroller.
- If the person is detected to have consumed alcohol neither wearing a seat belt nor helmet
- If the person is to be suspected as a violator, the face of the person is recognized using the face recognition IOT system. The crime is uploaded in the database.
- After recording the face and data related to the violator, the user can access whenever he/she wants using the fingerprint sensors.
- The data is also stored in the cloud network or in local backup server used by the police servers.
- These data contain the criminal violation, type of violation, suspect's personal information, date and time of crime occurred.
- These database are to be maintained in the cloud server provided by the police network and it can also be impenetrable.

HARDWARE COMPONENTS

- MICROCONTROLLER
- FINGER PRINT SENSOR
- UART
- DRIVER CIRCUIT
- DC MOTOR
- IOT
- GAS SENSOR
- LCD DISPLAY
- IR SENSOR
- POWER SUPPLY
- WSN
- SOFTWARE COMPONENTS:
- EMBEDDED C
- ARDUINO IDE
- PYTHON IDE

IV. ESP32 MICROCONTROLLER

you can make several projects either as a hobby or even commercial. As technology advanced, new project ideas and implementations came into play and one particular concept is the Internet of Things or IoT. It is a connected platform, where several

“things” or devices are connected over internet for exchange of information. In DIY community, the IOT projects are mainly focused on Home Automation and Smart Home applications but commercial and industrial IoT projects have far complex implementations like Machine Learning, Artificial Intelligence, Wireless Sensor Networks etc. The important thing in this brief intro is whether it is a small DIY project by a hobbyist or a complex industrial project, any IoT project must have connectivity to Internet. This is where the likes of ESP8266 and ESP32 come into picture. If you want to add Wi-Fi connectivity to your projects, then ESP8266 is a great option. But if you want build a complete system with Wi-Fi connectivity, Bluetooth connectivity, high resolution ADCs, DAC, Serial Connectivity and many other features, then ESP32 is the ultimate choice

#### V. FINGERPRINT READER WITH BOARD

SM-621 is RS232 /UART fingerprint module scanner for the demand of access control system, door lock, T&A and safety box OEM POS Consisting of high function DSP, large capacity FLASH and color CMOS, etc, SM621 optical fingerprint module can conduct fingerprint enrollment, image processing, templates storage, fingerprint matching and fingerprint searching. This Optical biometric fingerprint reader is with great features and can be embedded into a variety of end products, such as: access control, attendance, safety deposit box, car door locks. PRODUCT DESCRIPTION R305 fingerprint module is fingerprint sensor with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USBSerial adapter. The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port.

#### VI. DC MOTOR

The relationship between torque vs speed and current is linear as shown left; as the load on a motor increases, Speed will decrease. The graph pictured 26 here represents the characteristics of a typical motor. As long as the motor is used in the area of high efficiency (as represented by the shaded area) long life and good performance can be expected. However, using the motor outside this range will result in high temperature rises and deterioration of motor parts. A motor's basic rating point is slightly lower than its maximum efficiency point. Load torque can be determined by measuring the current drawn when the motor is attached to a machine whose actual load value is known. Geared

dc motors can be defined as an extension of dc motors. A geared DC Motor has a gear assembly attached to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM .The gear assembly helps in increasing the torque and reducing the speed. Using the correct combination of gears in a gear motor, its speed can be reduced to any desirable figure. This concept where gears reduce the speed of the vehicle but increase its torque is known as gear reduction. A DC motor can be used at a voltage lower than the rated voltage. But, below 1000 rpm, the speed becomes unstable, and the motor will not run smoothly.

#### VII. IOT

The Internet of things (IoT) is the network of everyday objects — physical things embedded with electronics, software, sensors, and connectivity enabling data exchange. Basically, a little networked computer is attached to a thing, allowing information exchange to and from that thing. Be it lightbulbs, toasters, refrigerators, flower pots, watches, fans, planes, trains, automobiles, or anything else around you, a little networked computer can be combined with it to accept input (especially object control) or to gather and generate informational output (typically object status or other sensory data). This means computers will be permeating everything around us — ubiquitous embedded computing devices, uniquely identifiable, interconnected across the Internet. Because of low-cost, networkable microcontroller modules, the Internet of things is really starting to take off.

#### VIII. WEB SERVER

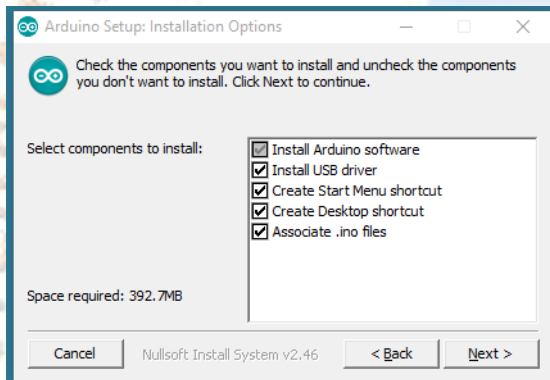
29 Espresso's ESP32 delivers highly integrated Wi-Fi SoC solution to meet users' continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry. With the complete and self-contained Wi-Fi networking capabilities, ESP32 can perform either as a standalone application or as the slave to a host MCU. When ESP32 hosts the application, it promptly boots up from the flash. The integrated high speed cache helps to increase the system performance and optimize the system memory. Also, ESP32 can be applied to any microcontroller design as a Wi-Fi adaptor through SPI / SDIO or I2C / UART interfaces. ESP32 integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. The compact design minimizes the PCB size and requires minimal external circuitries. Besides the Wi-Fi functionalities, ESP32 also integrates an enhanced version of Tensilica's L106



Diamond series 32-bit processor and on-chip SRAM. It can be interfaced with external sensors and other devices through the GPIOs. Software Development Kit (SDK) provides sample codes for various applications. Espressif Systems' Smart Connectivity Platform (ESCP) enables sophisticated features including fast switch between sleep and wakeup mode for energy-efficient purpose, adaptive radio biasing for low-power operation, advance signal processing, spur cancellation and radio coexistence mechanisms for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

## IX. ARDUINO SOFTWARE (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



The process will extract and install all the required files to execute properly the Arduino Software (IDE)

### Technical Details

The Arduino Boot loader sets the "erase Address" to zero every time the boot loader is called. ROBOTC called the "Load Address" command to set the address in which we want to write/verify when downloading program.

When writing a page of memory to the arduino, the Arduino boot loader will erase the existing page and write a whole new page.

In the scenario of downloading firmware, everything is great because the Erase Address and the Loaded Address both start at zero.

In the scenario of writing a user program, we start writing at memory location 0x7000, but the Boot loader erases information starting at location zero because the "Load Address" command doesn't update where to erase. Our modification is to set both the Load Address and the Erase Address so the activity of writing a user program doesn't cause the firmware to be accidentally erased.

*The power pins are as follows:*

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
  - **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via a non-board regulator, or be supplied by USB or another regulated 5V supply.
  - **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.
  - **GND.** Ground pins.
- The ATMEGA has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).
- Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50k Ohms. In addition, some pins have specialized functions:
- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATMEGA USB-to-TTL Serial chip.
  - **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a changing value. See the attachInterrupt() function for details.
  - **PWM:** 0 to 13. Provide 8-bit PWM output with the analogWrite() function.
  - **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
  - **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- I2C: 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove. The Arduino UNO has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and analog Reference() function. There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analog Reference().
- reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### PYTHON

Python is a widely used high-level programming language for generalpurpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a communitybased development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

```

1 # I2C
2
3 # I2C address
4 I2C_ADDR = 0x42
5
6 # I2C bus
7 I2C_BUS = 1
8
9 # I2C device name
10 I2C_DEVICE = "I2C Device"
11
12 # I2C device address
13 I2C_ADDR = 0x42
14
15 # I2C device name
16 I2C_DEVICE = "I2C Device"
17
18 # I2C device address
19 I2C_ADDR = 0x42
20
21 # I2C device name
22 I2C_DEVICE = "I2C Device"
23
24 # I2C device address
25 I2C_ADDR = 0x42
26
27 # I2C device name
28 I2C_DEVICE = "I2C Device"
29
30 # I2C device address
31 I2C_ADDR = 0x42
32
33 # I2C device name
34 I2C_DEVICE = "I2C Device"
35
36 # I2C device address
37 I2C_ADDR = 0x42
38
39 # I2C device name
40 I2C_DEVICE = "I2C Device"
41
42 # I2C device address
43 I2C_ADDR = 0x42
44
45 # I2C device name
46 I2C_DEVICE = "I2C Device"
47
48 # I2C device address
49 I2C_ADDR = 0x42
50
51 # I2C device name
52 I2C_DEVICE = "I2C Device"
53
54 # I2C device address
55 I2C_ADDR = 0x42
56
57 # I2C device name
58 I2C_DEVICE = "I2C Device"
59
60 # I2C device address
61 I2C_ADDR = 0x42
62
63 # I2C device name
64 I2C_DEVICE = "I2C Device"
65
66 # I2C device address
67 I2C_ADDR = 0x42
68
69 # I2C device name
70 I2C_DEVICE = "I2C Device"
71
72 # I2C device address
73 I2C_ADDR = 0x42
74
75 # I2C device name
76 I2C_DEVICE = "I2C Device"
77
78 # I2C device address
79 I2C_ADDR = 0x42
80
81 # I2C device name
82 I2C_DEVICE = "I2C Device"
83
84 # I2C device address
85 I2C_ADDR = 0x42
86
87 # I2C device name
88 I2C_DEVICE = "I2C Device"
89
90 # I2C device address
91 I2C_ADDR = 0x42
92
93 # I2C device name
94 I2C_DEVICE = "I2C Device"
95
96 # I2C device address
97 I2C_ADDR = 0x42
98
99 # I2C device name
100 I2C_DEVICE = "I2C Device"
101
102 # I2C device address
103 I2C_ADDR = 0x42
104
105 # I2C device name
106 I2C_DEVICE = "I2C Device"
107
108 # I2C device address
109 I2C_ADDR = 0x42
110
111 # I2C device name
112 I2C_DEVICE = "I2C Device"
113
114 # I2C device address
115 I2C_ADDR = 0x42
116
117 # I2C device name
118 I2C_DEVICE = "I2C Device"
119
120 # I2C device address
121 I2C_ADDR = 0x42
122
123 # I2C device name
124 I2C_DEVICE = "I2C Device"
125
126 # I2C device address
127 I2C_ADDR = 0x42
128
129 # I2C device name
130 I2C_DEVICE = "I2C Device"
131
132 # I2C device address
133 I2C_ADDR = 0x42
134
135 # I2C device name
136 I2C_DEVICE = "I2C Device"
137
138 # I2C device address
139 I2C_ADDR = 0x42
140
141 # I2C device name
142 I2C_DEVICE = "I2C Device"
143
144 # I2C device address
145 I2C_ADDR = 0x42
146
147 # I2C device name
148 I2C_DEVICE = "I2C Device"
149
150 # I2C device address
151 I2C_ADDR = 0x42
152
153 # I2C device name
154 I2C_DEVICE = "I2C Device"
155
156 # I2C device address
157 I2C_ADDR = 0x42
158
159 # I2C device name
160 I2C_DEVICE = "I2C Device"
161
162 # I2C device address
163 I2C_ADDR = 0x42
164
165 # I2C device name
166 I2C_DEVICE = "I2C Device"
167
168 # I2C device address
169 I2C_ADDR = 0x42
170
171 # I2C device name
172 I2C_DEVICE = "I2C Device"
173
174 # I2C device address
175 I2C_ADDR = 0x42
176
177 # I2C device name
178 I2C_DEVICE = "I2C Device"
179
180 # I2C device address
181 I2C_ADDR = 0x42
182
183 # I2C device name
184 I2C_DEVICE = "I2C Device"
185
186 # I2C device address
187 I2C_ADDR = 0x42
188
189 # I2C device name
190 I2C_DEVICE = "I2C Device"
191
192 # I2C device address
193 I2C_ADDR = 0x42
194
195 # I2C device name
196 I2C_DEVICE = "I2C Device"
197
198 # I2C device address
199 I2C_ADDR = 0x42
200
201 # I2C device name
202 I2C_DEVICE = "I2C Device"
203
204 # I2C device address
205 I2C_ADDR = 0x42
206
207 # I2C device name
208 I2C_DEVICE = "I2C Device"
209
210 # I2C device address
211 I2C_ADDR = 0x42
212
213 # I2C device name
214 I2C_DEVICE = "I2C Device"
215
216 # I2C device address
217 I2C_ADDR = 0x42
218
219 # I2C device name
220 I2C_DEVICE = "I2C Device"
221
222 # I2C device address
223 I2C_ADDR = 0x42
224
225 # I2C device name
226 I2C_DEVICE = "I2C Device"
227
228 # I2C device address
229 I2C_ADDR = 0x42
230
231 # I2C device name
232 I2C_DEVICE = "I2C Device"
233
234 # I2C device address
235 I2C_ADDR = 0x42
236
237 # I2C device name
238 I2C_DEVICE = "I2C Device"
239
240 # I2C device address
241 I2C_ADDR = 0x42
242
243 # I2C device name
244 I2C_DEVICE = "I2C Device"
245
246 # I2C device address
247 I2C_ADDR = 0x42
248
249 # I2C device name
250 I2C_DEVICE = "I2C Device"
251
252 # I2C device address
253 I2C_ADDR = 0x42
254
255 # I2C device name
256 I2C_DEVICE = "I2C Device"
257
258 # I2C device address
259 I2C_ADDR = 0x42
260
261 # I2C device name
262 I2C_DEVICE = "I2C Device"
263
264 # I2C device address
265 I2C_ADDR = 0x42
266
267 # I2C device name
268 I2C_DEVICE = "I2C Device"
269
270 # I2C device address
271 I2C_ADDR = 0x42
272
273 # I2C device name
274 I2C_DEVICE = "I2C Device"
275
276 # I2C device address
277 I2C_ADDR = 0x42
278
279 # I2C device name
280 I2C_DEVICE = "I2C Device"
281
282 # I2C device address
283 I2C_ADDR = 0x42
284
285 # I2C device name
286 I2C_DEVICE = "I2C Device"
287
288 # I2C device address
289 I2C_ADDR = 0x42
290
291 # I2C device name
292 I2C_DEVICE = "I2C Device"
293
294 # I2C device address
295 I2C_ADDR = 0x42
296
297 # I2C device name
298 I2C_DEVICE = "I2C Device"
299
300 # I2C device address
301 I2C_ADDR = 0x42
302
303 # I2C device name
304 I2C_DEVICE = "I2C Device"
305
306 # I2C device address
307 I2C_ADDR = 0x42
308
309 # I2C device name
310 I2C_DEVICE = "I2C Device"
311
312 # I2C device address
313 I2C_ADDR = 0x42
314
315 # I2C device name
316 I2C_DEVICE = "I2C Device"
317
318 # I2C device address
319 I2C_ADDR = 0x42
320
321 # I2C device name
322 I2C_DEVICE = "I2C Device"
323
324 # I2C device address
325 I2C_ADDR = 0x42
326
327 # I2C device name
328 I2C_DEVICE = "I2C Device"
329
330 # I2C device address
331 I2C_ADDR = 0x42
332
333 # I2C device name
334 I2C_DEVICE = "I2C Device"
335
336 # I2C device address
337 I2C_ADDR = 0x42
338
339 # I2C device name
340 I2C_DEVICE = "I2C Device"
341
342 # I2C device address
343 I2C_ADDR = 0x42
344
345 # I2C device name
346 I2C_DEVICE = "I2C Device"
347
348 # I2C device address
349 I2C_ADDR = 0x42
350
351 # I2C device name
352 I2C_DEVICE = "I2C Device"
353
354 # I2C device address
355 I2C_ADDR = 0x42
356
357 # I2C device name
358 I2C_DEVICE = "I2C Device"
359
360 # I2C device address
361 I2C_ADDR = 0x42
362
363 # I2C device name
364 I2C_DEVICE = "I2C Device"
365
366 # I2C device address
367 I2C_ADDR = 0x42
368
369 # I2C device name
370 I2C_DEVICE = "I2C Device"
371
372 # I2C device address
373 I2C_ADDR = 0x42
374
375 # I2C device name
376 I2C_DEVICE = "I2C Device"
377
378 # I2C device address
379 I2C_ADDR = 0x42
380
381 # I2C device name
382 I2C_DEVICE = "I2C Device"
383
384 # I2C device address
385 I2C_ADDR = 0x42
386
387 # I2C device name
388 I2C_DEVICE = "I2C Device"
389
390 # I2C device address
391 I2C_ADDR = 0x42
392
393 # I2C device name
394 I2C_DEVICE = "I2C Device"
395
396 # I2C device address
397 I2C_ADDR = 0x42
398
399 # I2C device name
400 I2C_DEVICE = "I2C Device"
401
402 # I2C device address
403 I2C_ADDR = 0x42
404
405 # I2C device name
406 I2C_DEVICE = "I2C Device"
407
408 # I2C device address
409 I2C_ADDR = 0x42
410
411 # I2C device name
412 I2C_DEVICE = "I2C Device"
413
414 # I2C device address
415 I2C_ADDR = 0x42
416
417 # I2C device name
418 I2C_DEVICE = "I2C Device"
419
420 # I2C device address
421 I2C_ADDR = 0x42
422
423 # I2C device name
424 I2C_DEVICE = "I2C Device"
425
426 # I2C device address
427 I2C_ADDR = 0x42
428
429 # I2C device name
430 I2C_DEVICE = "I2C Device"
431
432 # I2C device address
433 I2C_ADDR = 0x42
434
435 # I2C device name
436 I2C_DEVICE = "I2C Device"
437
438 # I2C device address
439 I2C_ADDR = 0x42
440
441 # I2C device name
442 I2C_DEVICE = "I2C Device"
443
444 # I2C device address
445 I2C_ADDR = 0x42
446
447 # I2C device name
448 I2C_DEVICE = "I2C Device"
449
450 # I2C device address
451 I2C_ADDR = 0x42
452
453 # I2C device name
454 I2C_DEVICE = "I2C Device"
455
456 # I2C device address
457 I2C_ADDR = 0x42
458
459 # I2C device name
460 I2C_DEVICE = "I2C Device"
461
462 # I2C device address
463 I2C_ADDR = 0x42
464
465 # I2C device name
466 I2C_DEVICE = "I2C Device"
467
468 # I2C device address
469 I2C_ADDR = 0x42
470
471 # I2C device name
472 I2C_DEVICE = "I2C Device"
473
474 # I2C device address
475 I2C_ADDR = 0x42
476
477 # I2C device name
478 I2C_DEVICE = "I2C Device"
479
480 # I2C device address
481 I2C_ADDR = 0x42
482
483 # I2C device name
484 I2C_DEVICE = "I2C Device"
485
486 # I2C device address
487 I2C_ADDR = 0x42
488
489 # I2C device name
490 I2C_DEVICE = "I2C Device"
491
492 # I2C device address
493 I2C_ADDR = 0x42
494
495 # I2C device name
496 I2C_DEVICE = "I2C Device"
497
498 # I2C device address
499 I2C_ADDR = 0x42
500
501 # I2C device name
502 I2C_DEVICE = "I2C Device"
503
504 # I2C device address
505 I2C_ADDR = 0x42
506
507 # I2C device name
508 I2C_DEVICE = "I2C Device"
509
510 # I2C device address
511 I2C_ADDR = 0x42
512
513 # I2C device name
514 I2C_DEVICE = "I2C Device"
515
516 # I2C device address
517 I2C_ADDR = 0x42
518
519 # I2C device name
520 I2C_DEVICE = "I2C Device"
521
522 # I2C device address
523 I2C_ADDR = 0x42
524
525 # I2C device name
526 I2C_DEVICE = "I2C Device"
527
528 # I2C device address
529 I2C_ADDR = 0x42
530
531 # I2C device name
532 I2C_DEVICE = "I2C Device"
533
534 # I2C device address
535 I2C_ADDR = 0x42
536
537 # I2C device name
538 I2C_DEVICE = "I2C Device"
539
540 # I2C device address
541 I2C_ADDR = 0x42
542
543 # I2C device name
544 I2C_DEVICE = "I2C Device"
545
546 # I2C device address
547 I2C_ADDR = 0x42
548
549 # I2C device name
550 I2C_DEVICE = "I2C Device"
551
552 # I2C device address
553 I2C_ADDR = 0x42
554
555 # I2C device name
556 I2C_DEVICE = "I2C Device"
557
558 # I2C device address
559 I2C_ADDR = 0x42
560
561 # I2C device name
562 I2C_DEVICE = "I2C Device"
563
564 # I2C device address
565 I2C_ADDR = 0x42
566
567 # I2C device name
568 I2C_DEVICE = "I2C Device"
569
570 # I2C device address
571 I2C_ADDR = 0x42
572
573 # I2C device name
574 I2C_DEVICE = "I2C Device"
575
576 # I2C device address
577 I2C_ADDR = 0x42
578
579 # I2C device name
580 I2C_DEVICE = "I2C Device"
581
582 # I2C device address
583 I2C_ADDR = 0x42
584
585 # I2C device name
586 I2C_DEVICE = "I2C Device"
587
588 # I2C device address
589 I2C_ADDR = 0x42
590
591 # I2C device name
592 I2C_DEVICE = "I2C Device"
593
594 # I2C device address
595 I2C_ADDR = 0x42
596
597 # I2C device name
598 I2C_DEVICE = "I2C Device"
599
600 # I2C device address
601 I2C_ADDR = 0x42
602
603 # I2C device name
604 I2C_DEVICE = "I2C Device"
605
606 # I2C device address
607 I2C_ADDR = 0x42
608
609 # I2C device name
610 I2C_DEVICE = "I2C Device"
611
612 # I2C device address
613 I2C_ADDR = 0x42
614
615 # I2C device name
616 I2C_DEVICE = "I2C Device"
617
618 # I2C device address
619 I2C_ADDR = 0x42
620
621 # I2C device name
622 I2C_DEVICE = "I2C Device"
623
624 # I2C device address
625 I2C_ADDR = 0x42
626
627 # I2C device name
628 I2C_DEVICE = "I2C Device"
629
630 # I2C device address
631 I2C_ADDR = 0x42
632
633 # I2C device name
634 I2C_DEVICE = "I2C Device"
635
636 # I2C device address
637 I2C_ADDR = 0x42
638
639 # I2C device name
640 I2C_DEVICE = "I2C Device"
641
642 # I2C device address
643 I2C_ADDR = 0x42
644
645 # I2C device name
646 I2C_DEVICE = "I2C Device"
647
648 # I2C device address
649 I2C_ADDR = 0x42
650
651 # I2C device name
652 I2C_DEVICE = "I2C Device"
653
654 # I2C device address
655 I2C_ADDR = 0x42
656
657 # I2C device name
658 I2C_DEVICE = "I2C Device"
659
660 # I2C device address
661 I2C_ADDR = 0x42
662
663 # I2C device name
664 I2C_DEVICE = "I2C Device"
665
666 # I2C device address
667 I2C_ADDR = 0x42
668
669 # I2C device name
670 I2C_DEVICE = "I2C Device"
671
672 # I2C device address
673 I2C_ADDR = 0x42
674
675 # I2C device name
676 I2C_DEVICE = "I2C Device"
677
678 # I2C device address
679 I2C_ADDR = 0x42
680
681 # I2C device name
682 I2C_DEVICE = "I2C Device"
683
684 # I2C device address
685 I2C_ADDR = 0x42
686
687 # I2C device name
688 I2C_DEVICE = "I2C Device"
689
690 # I2C device address
691 I2C_ADDR = 0x42
692
693 # I2C device name
694 I2C_DEVICE = "I2C Device"
695
696 # I2C device address
697 I2C_ADDR = 0x42
698
699 # I2C device name
700 I2C_DEVICE = "I2C Device"
701
702 # I2C device address
703 I2C_ADDR = 0x42
704
705 # I2C device name
706 I2C_DEVICE = "I2C Device"
707
708 # I2C device address
709 I2C_ADDR = 0x42
710
711 # I2C device name
712 I2C_DEVICE = "I2C Device"
713
714 # I2C device address
715 I2C_ADDR = 0x42
716
717 # I2C device name
718 I2C_DEVICE = "I2C Device"
719
720 # I2C device address
721 I2C_ADDR = 0x42
722
723 # I2C device name
724 I2C_DEVICE = "I2C Device"
725
726 # I2C device address
727 I2C_ADDR = 0x42
728
729 # I2C device name
730 I2C_DEVICE = "I2C Device"
731
732 # I2C device address
733 I2C_ADDR = 0x42
734
735 # I2C device name
736 I2C_DEVICE = "I2C Device"
737
738 # I2C device address
739 I2C_ADDR = 0x42
740
741 # I2C device name
742 I2C_DEVICE = "I2C Device"
743
744 # I2C device address
745 I2C_ADDR = 0x42
746
747 # I2C device name
748 I2C_DEVICE = "I2C Device"
749
750 # I2C device address
751 I2C_ADDR = 0x42
752
753 # I2C device name
754 I2C_DEVICE = "I2C Device"
755
756 # I2C device address
757 I2C_ADDR = 0x42
758
759 # I2C device name
760 I2C_DEVICE = "I2C Device"
761
762 # I2C device address
763 I2C_ADDR = 0x42
764
765 # I2C device name
766 I2C_DEVICE = "I2C Device"
767
768 # I2C device address
769 I2C_ADDR = 0x42
770
771 # I2C device name
772 I2C_DEVICE = "I2C Device"
773
774 # I2C device address
775 I2C_ADDR = 0x42
776
777 # I2C device name
778 I2C_DEVICE = "I2C Device"
779
780 # I2C device address
781 I2C_ADDR = 0x42
782
783 # I2C device name
784 I2C_DEVICE = "I2C Device"
785
786 # I2C device address
787 I2C_ADDR = 0x42
788
789 # I2C device name
790 I2C_DEVICE = "I2C Device"
791
792 # I2C device address
793 I2C_ADDR = 0x42
794
795 # I2C device name
796 I2C_DEVICE = "I2C Device"
797
798 # I2C device address
799 I2C_ADDR = 0x42
800
801 # I2C device name
802 I2C_DEVICE = "I2C Device"
803
804 # I2C device address
805 I2C_ADDR = 0x42
806
807 # I2C device name
808 I2C_DEVICE = "I2C Device"
809
810 # I2C device address
811 I2C_ADDR = 0x42
812
813 # I2C device name
814 I2C_DEVICE = "I2C Device"
815
816 # I2C device address
817 I2C_ADDR = 0x42
818
819 # I2C device name
820 I2C_DEVICE = "I2C Device"
821
822 # I2C device address
823 I2C_ADDR = 0x42
824
825 # I2C device name
826 I2C_DEVICE = "I2C Device"
827
828 # I2C device address
829 I2C_ADDR = 0x42
830
831 # I2C device name
832 I2C_DEVICE = "I2C Device"
833
834 # I2C device address
835 I2C_ADDR = 0x42
836
837 # I2C device name
838 I2C_DEVICE = "I2C Device"
839
840 # I2C device address
841 I2C_ADDR = 0x42
842
843 # I2C device name
844 I2C_DEVICE = "I2C Device"
845
846 # I2C device address
847 I2C_ADDR = 0x42
848
849 # I2C device name
850 I2C_DEVICE = "I2C Device"
851
852 # I2C device address
853 I2C_ADDR = 0x42
854
855 # I2C device name
856 I2C_DEVICE = "I2C Device"
857
858 # I2C device address
859 I2C_ADDR = 0x42
860
861 # I2C device name
862 I2C_DEVICE = "I2C Device"
863
864 # I2C device address
865 I2C_ADDR = 0x42
866
867 # I2C device name
868 I2C_DEVICE = "I2C Device"
869
870 # I2C device address
871 I2C_ADDR = 0x42
872
873 # I2C device name
874 I2C_DEVICE = "I2C Device"
875
876 # I2C device address
877 I2C_ADDR = 0x42
878
879 # I2C device name
880 I2C_DEVICE = "I2C Device"
881
882 # I2C device address
883 I2C_ADDR = 0x42
884
885 # I2C device name
886 I2C_DEVICE = "I2C Device"
887
888 # I2C device address
889 I2C_ADDR = 0x42
890
891 # I2C device name
892 I2C_DEVICE = "I2C Device"
893
894 # I2C device address
895 I2C_ADDR = 0x42
896
897 # I2C device name
898 I2C_DEVICE = "I2C Device"
899
900 # I2C device address
901 I2C_ADDR = 0x42
902
903 # I2C device name
904 I2C_DEVICE = "I2C Device"
905
906 # I2C device address
907 I2C_ADDR = 0x42
908
909 # I2C device name
910 I2C_DEVICE = "I2C Device"
911
912 # I2C device address
913 I2C_ADDR = 0x42
914
915 # I2C device name
916 I2C_DEVICE = "I2C Device"
917
918 # I2C device address
919 I2C_ADDR = 0x42
920
921 # I2C device name
922 I2C_DEVICE = "I2C Device"
923
924 # I2C device address
925 I2C_ADDR = 0x42
926
927 # I2C device name
928 I2C_DEVICE = "I2C Device"
929
930 # I2C device address
931 I2C_ADDR = 0x42
932
933 # I2C device name
934 I2C_DEVICE = "I2C Device"
935
936 # I2C device address
937 I2C_ADDR = 0x42
938
939 # I2C device name
940 I2C_DEVICE = "I2C Device"
941
942 # I2C device address
943 I2C_ADDR = 0x42
944
945 # I2C device name
946 I2C_DEVICE = "I2C Device"
947
948 # I2C device address
949 I2C_ADDR = 0x42
950
951 # I2C device name
952 I2C_DEVICE = "I2C Device"
953
954 # I2C device address
955 I2C_ADDR = 0x42
956
957 # I2C device name
958 I2C_DEVICE = "I2C Device"
959
960 # I2C device address
961 I2C_ADDR = 0x42
962
963 # I2C device name
964 I2C_DEVICE = "I2C Device"
965
966 # I2C device address
967 I2C_ADDR = 0x42
968
969 # I2C device name
970 I2C_DEVICE = "I2C Device"
971
972 # I2C device address
973 I2C_ADDR = 0x42
974
975 # I2C device name
976 I2C_DEVICE = "I2C Device"
977
978 # I2C device address
979 I2C_ADDR = 0x42
980
981 # I2C device name
982 I2C_DEVICE = "I2C Device"
983
984 # I2C device address
985 I2C_ADDR = 0x42
986
987 # I2C device name
988 I2C_DEVICE = "I2C Device"
989
990 # I2C device address
991 I2C_ADDR = 0x42
992
993 # I2C device name
994 I2C_DEVICE = "I2C Device"
995
996 # I2C device address
997 I2C_ADDR = 0x42
998
999 # I2C device name
1000 I2C_DEVICE = "I2C Device"

```

fig : 3 Code Snippet

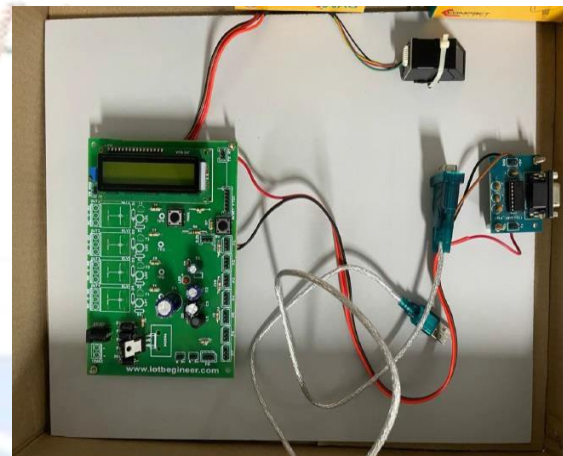


fig:4 System Prototype

### X. CONCLUSION

In the modern world, the use of computers and mobile phones is becoming rampant. As a result, the crime recording system needs to embrace the new technologies. This application will present a simple, convenient, cost-effective and efficient online crime recording system with a sensitive and intelligible web interface, thereby it reduces the amount of manual data entry. It is software which helps the policemen to work with the crimes and criminals easily. Additionally, it notifies the registered policemen about the release of criminals. This crime management system is a solution to all the problems related to the crime reports, criminal details and their crimes.

### XI. REFERENCES:

[1]. Oludeli Awodeli, Onulri Ernest. E, "A Real-Time Records Management System for National Security Agencies Department of Computer Science, Babcock University, Ilishan-Remo, Ogun State, Nigeria, Vol.3, Issued 12, May 2015.  
 [2]. Sourav Bhowmick, "Criminal Report Management System", Department of Computer Science and Engineering, ADMAS Institute of Technology, 2013.

- [3]. Mohammad Shahnawaz, "Crime Reporting and Crime Updates", 3rd International Conference on System Modeling in Research Trends (SMART) College of Computer Science and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, 2014.
- [4]. Srinidhi Eragam Reddy, Ramya Sahiti Amathi and Priyanka Vakkalagadda "Crime Reporting Interface Design using Mobile Technology", 2nd February, 2015.
- [5]. R. G. Jimoh, K. T. Ojulari, and O.A. Enikuomihin, "A Scalable Online Crime Reporting System", Department of Computer Science, University of Ilorin, Nigeria, Vol.7.No.1, January, 2014.
- [6] Highlights of 2009 Motor Vehicle crashes, Trac Safety Facts, Research Notes, NHTSA (National Highway traffic Safety Administration) [Online], Accessed on 16 October 2011.
- [7] N. Virtanen, A Schirokoff and I. Luom, "Impacts of an automatic emergency call system on accident consequences," 18th ICTCT, Workshop Transport telemetric and safety, 2005.
- [8] S. M. Tang and H. I. Gao, "Trac-incident detection- algorithm based on nonparametric regression," IEEE Transactions on Intelligent Transportation Systems, 2005.
- [9] G. Rose, "Mobile Phones as Traffic Probes: Practices, Prospects and Issues," Transport Reviews, 2006.
- [10] P.Mohan, Y.N.Padmanabhan, and R.Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in Proceedings of the 6th ACM conference on Embedded network sensor systems, 2008.
- [11] C. Thompson, I. White, B. Dougherty, A Albright and D.C. Schmidt, "Using Smartphones to Detect Car Accidents and Provide Situational Awareness to Emergency Responders," in Proceedings of 3'd Mobile Wireless Middleware, Operating Systems, and Applications Conference,
- [12] J. Yoon, B. Noble and M. Liu, "Surface street traffic estimation," in Proceedings of 5th International Conference on Mobile Systems, Applications, and Services, 2007.
- [13] Md. Syedul Amin, Jubayer Jalil, M. B. I. Reaz, "Accident Detection and Reporting System using GPS, GPRS and GSM Technology," in Proceedings of International Conference on Informatics, Electronics & Vision, 2012.
- [14] Chalernpol Saiprasert and Wasan Pattara-Atikom, "Smartphone Enabled Dangerous Driving Report System," in Proceedings of 46th Hawaii International Conference on System Sciences, 2013.