001 //Гиль Никита Александрович

002 //группа 150501

003 //Микропроцессорное устройство контроля

004 //параметров велосипеда

005

006 //библиотека для ИК приемника

007 #include <NecDecoder.h>

008 //библиотека для датчика температура

009 #include <Adafruit\_AHTX0.h>

010 //библиотека для lcd дисплея

011 #include <LiquidCrystal\_I2C.h>

012 //библиотека для gps датчика

013 #include <TinyGPS++.h>

014 //библиотека для порта передачи с gps датчиком

015 #include <SoftwareSerial.h>

016 //библиотека для часов реального времени

017 #include <iarduino\_RTC.h>

018 //библиотека для EEPROM

019 #include <EEPROM.h>

020

021 #define \_RX 0 //вывод RX ардуино

022 #define \_TX 1 //вывод TX ардуино

023 #define \_PIN\_13 13 //вывод D13 ардуино

024 #define \_PIN\_12 12 //вывод D12 ардуино

025 #define \_PIN\_11 11 //вывод D11 ардуино

026 #define \_PIN\_10 10 //вывод D10 ардуино

027 #define \_PIN\_8 8 //вывод D8 ардуино

028 #define \_PIN\_4 4 //вывод D4 ардуино

029 #define \_PIN\_3 3 //вывод D3 ардуино

030

031 #define \_INT0 0 //прерывание 0

032 #define \_INT1 1 //прерывание 1

033

034 //скорость передачи порта

035 #define speedGPSPort 9600

036

037 //количество стартовых сигналов

038 #define startSygnals 3

039 //стартовая задержка

040 #define startDelay 400

041

042 //стартовая скорость ограничения

043 #define startControlSpeed 25

044 //нижняя граница скорости ограничения

045 #define lowSpeed 10

046 //верхняя граница скорости ограничения

047 #define highSpeed 40

048

049 //размер колес

050 #define size26 "26"

051 #define size27\_5 "27.5"

052 #define size29 "29"

053

054 //длины окружности колес

055 #define length26 2.1

056 #define length27\_5 2.2

057 #define length29 2.3

058

059 //значения для вывода информации

060 #define time 0 //вывод времени

061 #define speedDir 1 //вывод скорости и направления

062 #define speedParams 2 //вывод параметров скорости

063 #define tempHum 3 //вывод температуры и влажности

064 #define distance 4 //вывод расстояний

065

066 //окно коллизий

067 #define windowCollision 80

068

069 //адреса данных в EEPROM

070 #define addressAllDist 0 //адрес всего расстояния

071 #define addressTodayDist 8 //адрес сегоднешнего расстояния

072 #define addressDay 16 //адрес дня последнего измерения

073 #define addressMonth 24 //адрес месяца последнего измерения

074

075 //задержки

076 #define delay1sec 1000 // 1 секунда

077 #define delay2sec 2000 // 2 секунда

078 #define delay4sec 4000 // 4 секунда

079

080 //частота пьезодинамика

081 #define freq\_2000 2000

082

083 //количество цифр в числе

084 #define have1Digit 10 //1 цифра

085 #define have2Digit 100 //2 цифры

086

087 //коды кнопок

088 //обнуление расстояния

089 #define button\_0 104

090 //настройка диагонали колеса

091 #define button\_1 48

092 //настройка диагонали колеса

093 #define button\_2 24

094 //настройка диагонали колеса

095 #define button\_3 122

096 //уменьшение контролируемой скорости

097 #define button\_minus 224

098 //увеличение контролируемой скорости

099 #define button\_plus 168

100 //контроль скорости

101 #define button\_pause 194

102 //переключение канала назад

103 #define button\_ch\_minus 2

104 //переключение канала вперед

105 #define button\_ch\_plus 34

106 //переключение изменения времени

107 #define button\_ch 98

108

109 //максимальный день

110 #define max\_day 31

111 //максимальный месяц

112 #define max\_month 12

113 //максимальные минуты

114 #define max\_minutes 59

115 //максимальные часы

116 #define max\_hours 23

117 //максимальный год

118 #define max\_year 2050

119 //минимальный год

120 #define min\_year 2000

121

122 //отключение изменения времени

123 #define change\_off 0

124 //изменение дня

125 #define change\_day 1

126 //изменение месяца

127 #define change\_month 2

128 //изменение года

129 #define change\_year 3

130 //изменение минут

131 #define change\_minutes 4

132 //изменение часов

133 #define change\_hours 5

134

135 //флаг времени

136 int flagTime = 0;

137

138 //структура времени

139 typedef struct timeStruct {

140 int hour; //часы

141 int min; //минуты

142 int sec; //секунды

143 };

144

145 int day; //текущий день

146 int month; //текущий месяц

147 int year; //текущий код

148 int minutes; //текущие минуты

149 int hours; //текущие часы

150

151 //структура для хранения времени

152 timeStruct tm;

153

154 //gps датчик

155 TinyGPSPlus gps;

156 //порт для соединения к gps

157 SoftwareSerial SoftSerial(\_RX, \_TX);

158

159 //lcd дисплей

160 LiquidCrystal\_I2C lcd(0x27, 20, 4);

161

162 //часы реального времени

163 iarduino\_RTC clock(RTC\_DS1302, \_PIN\_12, \_PIN\_8, \_PIN\_10);

164

165 //ИК приемник

166 NecDecoder ir;

167

168 //датчик температуры

169 Adafruit\_AHTX0 aht;

170 //значения измерений

171 sensors\_event\_t humidity, temp;

172

173 //скорость ограничения

174 int speedControl = startControlSpeed;

175 //флаг контроля скорости

176 volatile bool flagControlSpeed = true;

177 //скорость

178 int speedKmh;

179

180 //последнее измерение скорости

181 unsigned long lastFlash;

182

183 //последнее снятие значения с датчика температуры

184 unsigned long takeInfoAht;

185

186 //последняя запись в EEPROM и получение информации от gps

187 unsigned long writing;

188

189 //текущая длина окружности колеса

190 volatile float nowLength = length26;

191

192 //флаг для вывода информации

193 volatile int flagOut = 0;

194 //значение до изменения флага вывода

195 int lastFlagOut = 0;

196

197 //значение времени

198 unsigned long mill;

199

200 //расстояние за поездку

201 volatile double travelDist = 0;

202 //все расстояние

203 volatile double allDist = 0;

204 //расстояние за сегодня

205 volatile double todayDist = 0;

206

207 int currentDay; //день

208 int currentMonth; //месяц

209

210 //флаг очистки расстояний

211 bool flagClearDist = false;

212

213 //функция перевода секунд во время

214 void secondsToTime(unsigned long seconds) {

215 tm.sec = seconds % 60; //получение секунд

216 tm.min = (seconds / 60) % 60; //получение минут

217 tm.hour = (seconds / 3600) % 24; //получение часов

218 }

219

220 //функция обработки нажатия клавиш

221 void irIsr() {

222 ir.tick(); //начать обработку

223 //если доступно

224 if (ir.available()) {

225 //чтение значения кнопки

226 int value = ir.readCommand();

227

228 //если нажата кнопка паузы и установлен флаг вывода времени

229 if(value == button\_ch && flagOut == time) {

230 flagTime++;

231 if(flagTime > 5) {

232 flagTime = 0; //запрещение изменения времени

233 }

234 }

235

236 //если вывод времени

237 if(flagOut == time && flagTime != 0) {

238 //обновление времени

239 clock.gettime();

240 //получение текущего дня

241 day = (int)clock.day;

242 //получение текущего месяца

243 month = (int)clock.month;

244 //получение текущего года

245 year = (int)clock.year;

246 //получение текущих минут

247 minutes = (int)clock.minutes;

248 //получение текущих часов

249 hours = (int)clock.Hours;

250 //кнопка увеличения

251 if(value == button\_plus) {

252 //изменение дня

253 if(flagTime == 1) {

254 day++; //увеличение дня

255 if(day > max\_day) {

256 day = 1; //установка минимума

257 }

258 }

259 //изменение месяца

260 if(flagTime == 2) {

261 month++; //увеличение месяца

262 if(month > max\_month) {

263 month = 1; //установка минимума

264 }

265 }

266 //изменение года

267 if(flagTime == 3) {

268 year++; //увеличение года

269 if(year > max\_year) {

270 year = min\_year; //установка минимума

271 }

272 }

273 //изменение минут

274 if(flagTime == 4) {

275 minutes++; //увеличение минут

276 if(minutes > max\_minutes) {

277 //установка минимума

278 minutes = 0;

279 }

280 }

281 //изменение часов

282 if(flagTime == 5) {

283 hours++; //увеличение часов

284 if(hours > max\_hours) {

285 //установка минимума

286 hours = 0;

287 }

288 }

289 //установка нового времени

290 clock.settime(0, minutes, hours, day, month, year);

291 //кнопка уменьшения

292 } else if(value == button\_minus) {

293 //изменение дня

294 if(flagTime == 1) {

295 day--; //уменьшение дня

296 if(day < 1) {

297 day = max\_day; //установка максимума

298 }

299 }

300 //изменение месяца

301 if(flagTime == 2) {

302 month--; //уменьшение месяца

303 if(month < 1) {

304 month = max\_month;//установка максимума

305 }

306 }

307 //изменение года

308 if(flagTime == 3) {

309 year--; //уменьшение года

310 if(year < min\_year) {

311 year = max\_year; //установка максимума

312 }

313 }

314 //изменение минут

315 if(flagTime == 4) {

316 minutes--; //уменьшение минут

317 if(minutes < 0) {

318 //установка максимума

319 minutes = max\_minutes;

320 }

321 }

322 //изменение часов

323 if(flagTime == 5) {

324 hours--; //уменьшение часов

325 if(hours < 0) {

326 //установка максимума

327 hours = max\_hours;

328 }

329 }

330 //установка нового времени

331 clock.settime(0, minutes, hours, day, month, year);

332 }

333 }

334

335 //если нажата кнопка 1 и установлен флаг вывода параметров скорости

336 if(value == button\_1 && flagOut == speedParams) {

337 nowLength = length26; //изменить размер колеса

338 }

339 //если нажата кнопка 2 и установлен флаг вывода параметров скорости

340 if(value == button\_2 && flagOut == speedParams) {

341 nowLength = length27\_5; //изменить размер колеса

342 }

343 //если нажата кнопка 3 и установлен флаг вывода параметров скорости

344 if(value == button\_3 && flagOut == speedParams) {

345 nowLength = length29; //изменить размер колеса

346 }

347

348 //если нажата кнопка минус, контролируемая скорость больше минимальной

349 // и установлен флаг вывода параметров скорости

350 if(value == button\_minus && speedControl > lowSpeed

351 && flagOut == speedParams) {

352 speedControl--; //уменьшение контролируемой скорости

353 }

354 //если нажата кнопка плюс, контролируемая скорость меньше максимальной

355 // и установлен флаг вывода параметров скорости

356 if(value == button\_plus && speedControl < highSpeed

357 && flagOut == speedParams) {

358 speedControl++; //увеличение контролируемой скорости

359 }

360

361 //если нажата кнопка переключение канала назад

362 if(value == button\_ch\_minus) {

363 flagOut--;

364 if(flagOut < 0) {

365 flagOut = 4;

366 }

367 }

368

369 //если нажата кнопка переключение канала вперед

370 if(value == button\_ch\_plus) {

371 flagOut++;

372 if(flagOut > 4) {

373 flagOut = 0;

374 }

375 }

376

377 //если нажата кнопка 0 и флаг вывода установлен на расстояние

378 if(value == button\_0 && flagOut == distance) {

379 //обнуление всех расстояний

380 allDist = 0;

381 todayDist = 0;

382 travelDist = 0;

383 //установка флага обнуления расстояний

384 flagClearDist = true;

385 }

386

387 //если нажата кнопка пауза и установлен флаг вывода

388 //параметров скорости

389 if(value == button\_pause && flagOut == speedParams) {

390 if(flagControlSpeed){

391 flagControlSpeed = false; //сброс флага котроля скорости

392 } else {

393 flagControlSpeed = true; //установка флага котроля скорости

394 }

395 }

396 }

397 }

398

399 //функция измерения скорости

400 void speed() {

401 //сохранение времени измерения

402 mill = millis();

403 //ожидание окна коллизий

404 if(mill - lastFlash > windowCollision){

405 //вычисление скорости

406 speedKmh = (nowLength / (((float)(mill - lastFlash)) / 1000)) \* 3.6;

407 //сохранение последнего времени измерения

408 lastFlash = mill;

409 //увеличение расстояний

410 travelDist += nowLength;

411 allDist += nowLength;

412 todayDist += nowLength;

413 }

414 }

415

416 //функция вывода температуры и влажности

417 void outTempHum() {

418 //установка курсора в позицию 0 0

419 lcd.setCursor(0, 0);

420 //вывод температуры

421 lcd.print("Temperature: ");

422 lcd.print((int)temp.temperature);

423 lcd.print(" C");

424 //установка курсора в позицию 1 1

425 lcd.setCursor(0, 1);

426 //вывод влажности

427 lcd.print("Humidity: ");

428 lcd.print((int)humidity.relative\_humidity);

429 lcd.print(" %");

430 }

431

432 //функция вывода скорости и направления

433 void outSpeedDir() {

434 //установка курсора в позицию 0 0

435 lcd.setCursor(0, 0);

436 //вывод скорости

437 lcd.print("SPEED (km/h): ");

438 lcd.print(speedKmh);

439 lcd.print(" ");

440 //установка курсора в позицию 0 1

441 lcd.setCursor(0, 1);

442 lcd.print("DIRACTION");

443 //установка курсора в позицию 0 2

444 lcd.setCursor(0, 2);

445 lcd.print("degrees: ");

446 //если курс определен

447 if(gps.course.isValid()) {

448 //получение градусов

449 int deg = gps.course.deg();

450 //вывод градусов

451 lcd.print(deg);

452 //если цифр меньше 2

453 if(deg < have2Digit) {

454 lcd.print(" "); //удаление оставшейся цифры

455 }

456 //если цифр меньше 1

457 if(deg < have1Digit) {

458 lcd.print(" "); //удаление оставшейся цифры

459 }

460 }

461 }

462

463 //функция вывод параметров скорости

464 void outSpeedParams() {

465 //установка курсора в позицию 0 0

466 lcd.setCursor(0, 0);

467 lcd.print("SIZE: ");

468 //если размер 26 дюймов

469 if(nowLength == length26){

470 lcd.print(size26); //вывод размера

471 lcd.print(" ");

472 //если размер 27.5 дюймов

473 } else if(nowLength == length27\_5) {

474 lcd.print(size27\_5); //вывод размера

475 //если размер 29 дюймов

476 } else {

477 lcd.print(size29); //вывод размера

478 lcd.print(" ");

479 }

480 //установка курсора в позицию 0 1

481 lcd.setCursor(0, 1);

482 lcd.print("CONTROL: ");

483 //если скорость контролируется

484 if(flagControlSpeed) {

485 lcd.print("ON"); //вывод сообщения "включено"

486 lcd.print(" ");

487 //если скорость не контролируется

488 } else {

489 lcd.print("OFF"); //вывод сообщения "отключено"

490 }

491 //установка курсора в позицию 0 2

492 lcd.setCursor(0, 2);

493 //вывод контролируемой скорости

494 lcd.print("CONTROL SPEED: ");

495 lcd.print(speedControl);

496 }

497

498 //функция вывода времени

499 void outTime() {

500 //установка курсора в позицию 0 0

501 lcd.setCursor(0, 0);

502 //вывод текущей даты

503 lcd.print("DATE: ");

504 lcd.print(clock.gettime("d-m-Y"));

505 //установка курсора в позицию 0 1

506 lcd.setCursor(0, 1);

507 //вывод текущего времени

508 lcd.print("TIME: ");

509 lcd.print(clock.gettime("H:i:s"));

510 //установка курсора в позицию 0 2

511 lcd.setCursor(0, 2);

512 //вывод времени поездки

513 lcd.print("TRAVEL: ");

514 secondsToTime(millis() / 1000);

515 //если число состоит 1 цифры

516 if(tm.hour < have1Digit) {

517 lcd.print("0");

518 }

519 //вывод часов

520 lcd.print(tm.hour);

521 lcd.print(":");

522 //если число состоит 1 цифры

523 if(tm.min < have1Digit) {

524 lcd.print("0");

525 }

526 //вывод минут

527 lcd.print(tm.min);

528 lcd.print(":");

529 //если число состоит 1 цифры

530 if(tm.sec < have1Digit) {

531 lcd.print("0");

532 }

533 //вывод секунд

534 lcd.print(tm.sec);

535 //установка курсора в позицию 0 3

536 lcd.setCursor(0, 3);

537 lcd.print("CHANGE: ");

538 //ничего не изменяемо в дате

539 if(flagTime == change\_off) {

540 lcd.print("OFF"); //вывод сообщения "выключено"

541 lcd.print(" ");

542 } else if(flagTime == change\_day) {

543 lcd.print("DAY"); //изменение дня

544 lcd.print(" ");

545 } else if(flagTime == change\_month) {

546 lcd.print("MONTH"); //изменение месяца

547 lcd.print(" ");

548 } else if(flagTime == change\_year) {

549 lcd.print("YEAR"); //изменение года

550 lcd.print(" ");

551 } else if(flagTime == change\_minutes) {

552 lcd.print("MINUTES"); //изменение минут

553 } else if(flagTime == change\_hours) {

554 lcd.print("HOURS"); //изменение часов

555 lcd.print(" ");

556 }

557 }

558

559 //вывод расстояний

560 void outDistance() {

561 //если установлен флаг очистки

562 if(flagClearDist) {

563 flagClearDist = false; //сброс флага очистки

564 lcd.clear(); //очистка экрана

565 }

566 //установка курсора в позицию 0 0

567 lcd.setCursor(0, 0);

568 lcd.print("DISTANCE (m)");

569 //установка курсора в позицию 0 1

570 lcd.setCursor(0, 1);

571 //вывод расстояния за все время

572 lcd.print("ALL TIME: ");

573 lcd.print((long)allDist);

574 //установка курсора в позицию 0 2

575 lcd.setCursor(0, 2);

576 //вывод расстояния за сегодня

577 lcd.print("TODAY: ");

578 lcd.print((long)todayDist);

579 //установка курсора в позицию 0 3

580 lcd.setCursor(0, 3);

581 //вывод расстояния за поездку

582 lcd.print("NOW: ");

583 lcd.print((long)travelDist);

584 }

585

586 //установка начальных значений

587 void setup() {

588 //включение порта GPS

589 SoftSerial.begin(speedGPSPort);

590

591 //установка выводов на выход

592 pinMode(\_PIN\_13, OUTPUT);

593 pinMode(\_PIN\_11, OUTPUT);

594 pinMode(\_PIN\_4, OUTPUT);

595

596 //установка 0 прерывания (ик приемник)

597 attachInterrupt(\_INT0, irIsr, FALLING);

598

599 //установка 1 прерывания (герконовый датчик)

600 attachInterrupt(\_INT1, speed, RISING);

601

602 //запуск работы с датчиком температуры

603 aht.begin();

604

605 //запуск дисплея

606 lcd.init();

607 lcd.backlight();

608

609 //цикл сигнализирования о запуске устройства

610 for(int i = 0; i < startSygnals; ++i){

611 //включение диода

612 digitalWrite(\_PIN\_13, HIGH);

613 //задержка

614 delay(startDelay);

615 //выключение диода

616 digitalWrite(\_PIN\_13, LOW);

617 //задержка

618 delay(startDelay);

619 }

620

621 //запуск работы с часами реального времени

622 clock.begin();

623 //получение из памяти всего пути

624 EEPROM.get(addressAllDist, allDist);

625 //получение из памяти пути за день

626 EEPROM.get(addressTodayDist, todayDist);

627 //получение дня последнего запуска

628 EEPROM.get(addressDay, currentDay);

629 //получение месяца последнего запуска

630 EEPROM.get(addressMonth, currentMonth);

631 //если дни не совпадают

632 if(currentDay != clock.day || currentMonth != clock.month) {

633 //обнуление расстояния за сегодня

634 todayDist = 0;

635 //запись в память расстояния, дня и месяца

636 EEPROM.put(addressTodayDist, todayDist);

637 EEPROM.put(addressDay, (int)clock.day);

638 EEPROM.put(addressMonth, (int)clock.month);

639 }

640 }

641

642 //цикл выполнения программы

643 void loop() {

644 //если пришло время записи в память и чтения из порта gps

645 if(millis() - writing > delay1sec) {

646 writing = millis(); //сохрание времени последней записи

647 //сохранение всего расстояния

648 EEPROM.put(addressAllDist, allDist);

649 //сохранение расстояния за сегодня

650 EEPROM.put(addressTodayDist, todayDist);

651 //если порт доступен

652 if(SoftSerial.available() > 0) {

653 //декодирование данных

654 gps.encode(SoftSerial.read());

655 }

656 }

657

658 //если долгое время не было измерений

659 if (millis() - lastFlash > delay2sec) {

660 speedKmh = 0; //обнуление скорости

661 }

662

663 //если пришло время получить вданные с датчика температуры

664 if(millis() - takeInfoAht > delay4sec) {

665 //сохранение времени обращения к датчику температуры

666 takeInfoAht = millis();

667 //получение значений с датчика

668 aht.getEvent(&humidity, &temp);

669 }

670

671 //фиксация превышения скорости

672 if(speedKmh > speedControl && flagControlSpeed) {

673 digitalWrite(\_PIN\_11, HIGH); //открытие реле

674 tone(\_PIN\_4, freq\_2000); //запустили звучание

675 } else {

676 digitalWrite(\_PIN\_11, LOW); //закрытие реле

677 noTone(\_PIN\_4); //остановили звучание

678 }

679

680 //если флаг вывода был изменен

681 if(flagOut != lastFlagOut) {

682 lcd.clear(); //очистка экрана

683 //сохранение последнего значения флага

684 lastFlagOut = flagOut;

685 }

686

687 //если флаг установлен в вывод параметров скорости

688 if(flagOut == speedParams) {

689 outSpeedParams(); //вывод параметров скорости

690 }

691

692 //если флаг установлен в вывод скорости и направления

693 if(flagOut == speedDir) {

694 outSpeedDir(); //вывод скорости и направления

695 }

696

697 //если флаг установлен в вывод температуры и влажности

698 if(flagOut == tempHum) {

699 outTempHum(); //вывод температуры и влажности

700 }

701

702 //если флаг установлен в вывод времени

703 if(flagOut == time) {

704 outTime(); //вывод времени

705 }

706

707 //если флаг установлен в вывод расстояния

708 if(flagOut == distance) {

709 outDistance(); //вывод расстояния

710 }

711 }

712