Relatório Projeto Digital Experience Ultimate

Alunos: Luciano Ventura, Eduardo Manente

1. Sobre o projeto

O Sistema Inteligente de Segurança e Monitoramento Residencial foi desenvolvido para aumentar a segurança, o conforto e o monitoramento ambiental em uma residência.

Ele combina sensores (para detectar movimento, luz, temperatura, presença de intrusos e sensor infravermelho) com atuadores (LEDs, buzzer, display LCD e controle remoto) para agir automaticamente e informar o morador.

2. Funcionamento do projeto

1. O sensor LDR mede a luz ambiente, o sensor PIR detecta movimento.
2. Se for escuro e houver movimento, o sistema acende automaticamente o LED de iluminação.
3. Quando o movimento cessa, a luz se apaga após alguns segundos.
4. A temperatura é lida periodicamente e mostrada no LCD.
5. O sensor ultrassônico (HC-SR04) mede a distância à frente dele.
6. Se algo se aproximar a menos de 30 cm, o sistema considera intrusão e aciona o alarme (buzzer + LED vermelho + mensagem no LCD).
7. O controle remoto ativa ou desativa o alarme.
8. O alarme toca por alguns segundos e depois se desliga automaticamente.

3. Componentes do projeto

3.1 Sensores

- **Sensor PIR (Movimento):** Os sensores infravermelhos passivos (PIR) detectam energia térmica comparando os sinais de um par de elementos piroelétricos. O circuito de suporte envia então um sinal ALTO para o fio de sinal correspondente. Responsável por detectar a presença de movimento em seu campo de visão. Quando acionado, envia um sinal que é utilizado para acender a luz e indicar a presença de alguém no ambiente.
- **Sensor LDR (Luminosidade):** Quando a energia luminosa entra em contato com o sensor, o material absorve parte dessa energia e a converte em energia elétrica, tornando-se menos resistivo. Mede o nível de luminosidade do ambiente, permitindo que o sistema identifique se está escuro. Atua em conjunto com o sensor PIR para garantir que a luz seja acionada apenas em condições de baixa iluminação.
- **Sensor TMP36 (Temperatura):** Este dispositivo contém material semicondutor cujas propriedades elétricas se alteram de forma consistente com a variação de

temperatura. O sinal do pino central muda conforme a temperatura varia, e o programa do microcontrolador pode converter esse sinal de 0 a 1023 para uma temperatura em °C ou °F e é exibido no display LCD.

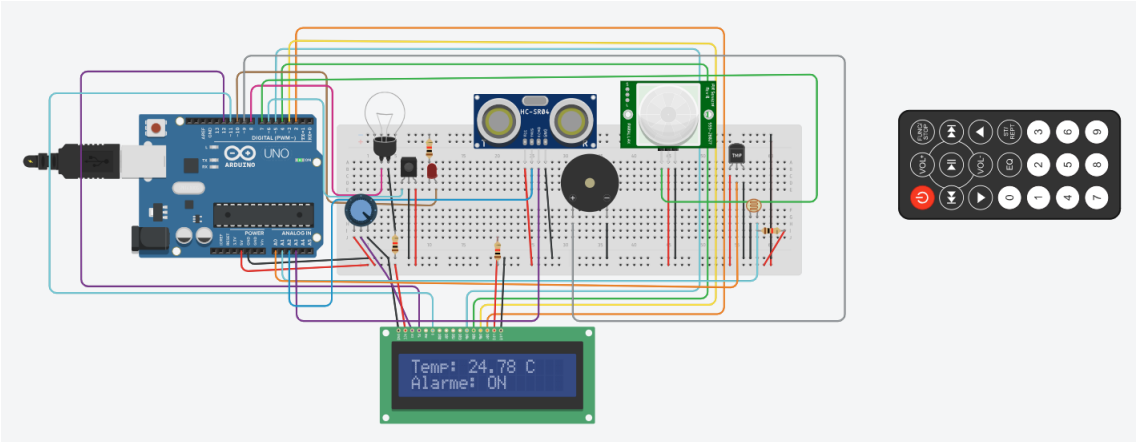
- **Sensor Ultrassônico HC-SR04:** Este dispositivo emite um som com uma frequência muito alta. Tão alta, na verdade, que você não consegue ouvi-la. O som leva tempo para se propagar pelo ar. Este dispositivo inteligente detecta o primeiro eco que reflete em um objeto próximo. Em seguida, ele calcula a distância do objeto medindo o tempo que o som leva para refletir no alvo e retornar. Utilizado para medir a distância de objetos próximos. Quando o sensor identifica a presença de um objeto ou pessoa a uma distância inferior a 30 cm, o sistema interpreta a situação como uma possível intrusão e aciona o alarme.
- **Sensor Infravermelho (IR):** Utilizado com um controle remoto infravermelho, um sensor infravermelho detecta padrões de luz para comunicação sem fio. Assim recebendo sinais do controle remoto e permitindo a ativação ou desativação do sistema de alarme de forma remota.
- **Potenciômetro:** Um potenciômetro funciona alterando a posição de um contato móvel em um material resistivo em relação a um contato fixo, resultando em uma variação da resistência conforme o contato é movido. Permite ajustar o contraste do display LCD, variando a tensão aplicada no terminal V0, de modo que a tela possa ser configurada para maior ou menor luminosidade, conforme a necessidade.

3.2 Atuadores

- **Buzzer (Emissor Sonoro):** O buzzer piezoelétrico vibra quando recebe corrente alternada, produzindo som. No sistema, ele emite um alerta sonoro quando o alarme é disparado.
- **Display LCD 16x2:** Os LCDs contêm muitas camadas de materiais. Há uma luz de fundo LED, bem como uma camada de vidro polarizado em torno de cristais líquidos, que pode girar eletronicamente a luz polarizada para permitir que a luz de fundo passe e seja vista, ou seja, bloqueada pelo polarizador na camada superior. Assim exibindo informações relevantes do sistema, como a temperatura ambiente, o status do alarme e mensagens de aviso ou alerta.
- **Lâmpada de LED:** Este dispositivo emite luz quando recebe eletricidade, aquecendo um filamento metálico a uma temperatura que produz luz. Simula a iluminação principal de um ambiente doméstico, sendo acionada automaticamente quando o sensor PIR detecta movimento em condições de pouca luz.
- **LED Vermelho (Indicador de Emergência):** Os LEDs são feitos de materiais semicondutores que emitem luz quando a corrente elétrica flui através deles na direção correta. Corrente excessiva pode danificá-los ou quebrá-los, portanto, é necessário adicionar um resistor em série para limitar a corrente. No sistema, atua como sinal visual de alerta, acendendo simultaneamente ao buzzer em situações de intrusão.

- **Controle Remoto:** Utilizado com um sensor infravermelho, um controle remoto infravermelho emite padrões de luz para comunicação sem fio. Permite ao usuário ativar ou desativar o sistema de alarme à distância, por meio da comunicação com o sensor infravermelho.

4. Hardware do projeto



5. Código do projeto

```
#include <LiquidCrystal.h>
#include <IRremote.hpp> // Biblioteca IR para controle remoto

 LCD de cristal líquido ( 12 , 11 , 5 , 4 , 3 , 2 );

// Pinos dos sensores e atuadores
#define PIN_PIR 7
#define PIN_TMP36 A0
#definir PIN_LDR A1
#define PIN_LED_LIGHT 8
#define PIN_BUZZER 9
#define PIN_ALARM_LED 10
#define PIN_TRIGGER A2
#define PIN_ECHO A3
#define PIN_IR 6 // Receptor IR

// Configurações
const  unsigned  long  TEMP_READ_INTERVAL_MS  =  5000UL ;   // 5 segundos
const  não assinado  longo  INTRUSION_MSG_INTERVAL  =  2000UL ; // Intervalo de mensagens de intrusão
const  int  LIGHT_THRESHOLD  =  400 ;   // Limiar de luminosidade
const  float  TEMP_HIGH_C  =  38,0 ;   // Temperatura alta de alerta
const  float  TEMP_LOW_C  =  35,0 ;    // Temperatura baixa de alerta
const  int  DIST_INTRUSAO_CM  =  30 ; //Distância limite para intrusão
const  não assinado  longo  IR_CODE_TOGGLE  =  0xFF00BF00 ;   // Código IR para alternar o alarme

// Variáveis de estado
bool  lightOn  =  false ;
bool  alarmArmed  =  verdadeiro ;
bool  alarmActive  =  false ;
bool  intrusaoDetectada  =  false ;

int  lastPirState  =  LOW ;
unsigned  long  lastTempReadMillis  =  0 ;
unsigned  long  lastIntrusionMsgMillis  =  0 ;
```

```
// gui do buzzer manual
 buzzerInterval longo  não assinado = 200 ;   //Intervalo de som do buzzer em ms
não assinado  longo  lastBuzzerMillis  =  0 ;    //Tempo de controle do buzzer

 configuração vazia () {
  pinMode ( PIN_PIR , INPUT );
  pinMode ( PIN_LED_LIGHT , OUTPUT );
  pinMode ( PIN_BUZZER , OUTPUT );
  pinMode ( PIN_ALARM_LED , OUTPUT );
  pinMode ( PIN_TRIGGER , OUTPUT );
  pinMode ( PIN_ECHO , INPUT );

  digitalWrite ( PIN_LED_LIGHT , LOW );
  digitalWrite ( PIN_BUZZER , LOW );
  digitalWrite ( PIN_ALARM_LED , LOW );

  Serial.begin ( 9600 ) ;
  lcd.begin ( 16 , 2 ) ;
  lcd.limpar ( ) ;

  // Inicializar o receptor IR
  IrReceptor . começar ( PIN_IR , 0 ); // Desabilita o feedback do LED do receptor IR

  //Mostra temperatura e estado do alarme no LCD
  float  tempInicial  =  readTemperatureC_TMP36 ();
  lcd.setCursor ( 0 , 0 ) ;
  lcd.print ( "Temp: " ) ;
  lcd.print ( tempInicial ) ;
  lcd.print ( " C " ) ;
  lcd.setCursor ( 0 , 1 ) ;
  lcd.print ( "Alarme: " ) ;
  lcd.print ( alarmArmed ? "ON" : " OFF " );

  Série . println ( "[Sistema] Inicializado." );
  Série . print ( "[Temp inicial] " );
  Serial.print ( tempInicial ) ;
  Serial.println ( " C" ) ;
  atraso ( 2000 );
}
```

```
void  loop () {
  unsigned  long  agora  =  millis ();

  // --- Controle IR ---
  se ( IrReceiver . decodificar ()) {
    unsigned  long  code  =  IrReceiver.decodedIRData.decodedRawData ;

    se ( código  ==  IR_CODE_TOGGLE ) {
      alarmeArmado  =  ! alarmeArmado ;
      se ( alarmeArmado ) {
        Série . println ( "[Controle] Alarme ativado!" );
      } outro {
        Série . println ( "[Controle] Alarme desativado!" );
        pararAlarme (); // Desativa o alarme quando o controle remoto é pressionado
      }
      lcd.setCursor ( 0 , 1 ) ;
      lcd.print ( "Alarme: " ) ;
      lcd.print ( alarmArmed ? "ON" : " OFF " );
      lcd.print ( " " ) ;
    }
    IrReceptor . retomar (); // Redefinir o IR para o próximo código
    atraso ( 50 );   // Pausa pequena para evitar sobrecarga no processo
  }

  // --- LDR (luminosidade) ---
  int  ldrVal  =  analogRead ( PIN_LDR );
  bool  claro  =  isDark ( ldrVal ); // Agora retorna true quando está claro

  // --- PIR (movimento) ---
  int  pirState  =  digitalRead ( PIN_PIR );

  se ( pirState  ==  ALTO  &&  últimoPirState  ==  BAIXO ) {
    Série . println ( "[PIR] Movimento detectado." );
    if ( claro  &&  ! lightOn ) turnLightOn ();   // Acende a luz se estiver claro
  }
  se ( pirState  ==  LOW  &&  lastPirState  ==  HIGH ) {
    Série . println ( "[PIR] Movimento cessou." );
    if ( claro  &&  lightOn ) turnLightOff ();   //Desliga a luz se estiver claro
  }
  últimoPirState  =  pirState ;
```

```
  // --- Temperatura ---
  se ( agora  -  últimaTempReadMillis  >=  TEMP_READ_INTERVAL_MS ) {
    lastTempReadMillis  =  agora ;
    float  temp  =  readTemperatureC_TMP36 ();
    Serial.print ( "[ Temp ] " ) ;
    Serial.print ( temp ) ;
    Serial.println ( " C" ) ;
    lcd.setCursor ( 0 , 0 ) ;
    lcd.print ( "Temp: " ) ;
    lcd.print ( temp ) ;
    lcd.print ( " C " ) ;
  }

  // --- Ultrassom (detecção de intrusão) ---
   distância  longa =  medirDistanciaCM ();

  if ( alarmeArmed  &&  distancia  >  0  &&  distancia  <=  DIST_INTRUSAO_CM ) {
    se ( ! intrusãoDetectada ) {
      intrusaoDetectada  =  true ;
      Série . println ( "[HC-SR04] Intrusão detectada!" );
      triggerAlarm ( "Intrusão detectada!" );
      lastIntrusionMsgMillis  =  agora ;
    } else  if ( now  -  lastIntrusionMsgMillis  >=  INTRUSION_MSG_INTERVAL ) {
      Série . println ( "[HC-SR04] Intrusão ainda presente!" );
      lastIntrusionMsgMillis  =  agora ;
    }
  } else  if ( intrusaoDetectada  && ( dista  >  DIST_INTRUSAO_CM  ||  distancia  <  0 )) {
    Série . println ( "[HC-SR04] Área limpa, alarme desativado." );
    intrusaoDetectada  =  false ;
    pararAlarme ();
  }

  // --- Controle do buzzer sem interrupção ---
  se ( alarmeAtivo ) {
    se ( agora  -  últimoBuzzerMillis  >=  buzzerInterval ) {
      últimoBuzzerMillis  =  agora ;
      // Alterna o estado do buzzer
      digitalWrite ( PIN_BUZZER , ! digitalRead ( PIN_BUZZER ));
    }
  }
}

// -------- Luz --------
bool  isDark ( int  ldrReading ) {
  return ( ldrReading  >  LIGHT_THRESHOLD ); //Retorna true quando está claro
}

void  ligarLuz () {
  digitalWrite ( PIN_LED_LIGHT , HIGH );
  luzLigada  =  verdadeiro ;
  Série . println ( "[Luz] Ligada (movimento + claro)." );
}

void  desligarLuz () {
  digitalWrite ( PIN_LED_LIGHT , LOW );
  luzLigada  =  falso ;
  Série . println ( "[Luz] Desligada (sem movimento + claro)." );
}

// -------- Alarme --------
void  triggerAlarm ( const  char  * reason ) {
  if ( ! alarmeArmed ) retornar ;   // Só ativa o alarme se estiver armado
  alarmActive  =  true ;
  Série . imprimir ( "[ALARME]" );
  Serial.println ( motivo ) ;
  digitalWrite ( PIN_ALARM_LED , ALTO );   // Acende o LED do alarme
  lcd.setCursor ( 0 , 1 ) ;
  lcd . print ( "** ALERTA!!! **" );
}

void  stopAlarm () {
  alarmActive  =  false ;
  digitalWrite ( PIN_BUZZER , BAIXO );   //Desliga a campainha
  digitalWrite ( PIN_ALARM_LED , BAIXO );   // Apaga o LED do alarme
  Série . println ( "[Alarme] Desligado." );
  lcd.setCursor ( 0 , 1 ) ;
  lcd.print ( "Alarme: " ) ;
  lcd.print ( alarmArmed ? "ON" : " OFF " );
  lcd.print ( " " ) ;
}
```

```
// -------- Temperatura --------
float  readTemperatureC_TMP36 () {
  int  raw  =  analogRead ( PIN_TMP36 );
   tensão  flutuante =  bruto  * ( 5,0  /  1023,0 );
  float  tempC  = ( voltage  -  0.5 ) *  100.0 ;
  retornar  tempC ;
}

// -------- Ultrassonografia --------
long  medirDistanciaCM () {
  digitalWrite ( PIN_TRIGGER , LOW );
  atrasoMicrosegundos ( 2 );
  digitalWrite ( PIN_TRIGGER , HIGH );
  atrasoMicrosegundos ( 10 );
  digitalWrite ( PIN_TRIGGER , LOW );

  longa  duração  =  pulseIn ( PIN_ECHO , HIGH , 25000 );
  se ( duracao  ==  0 ) retorne  -1 ;

   distância  longa =  duração  *  0,0343  /  2 ;
   distância de retorno ;
}
```

6. Monitor Serial

Serial Monitor

```
[Sistema] Inicializado.
[Temp inicial] 24.78 C
[Temp] 17.94 C
[PIR] Movimento detectado.
[Luz] Ligada (movimento + claro).
[PIR] Movimento cessou.
[Luz] Desligada (sem movimento + claro).
[PIR] Movimento detectado.
[PIR] Movimento cessou.
[Temp] 17.94 C
[Sistema] Inicializado.
[Temp inicial] 24.78 C
[Controle] Alarme desativado!
[Alarme] Desligado.
[Controle] Alarme ativado!
```

7. Benefícios e aplicações

- **Segurança:** Detecta movimentos suspeitos e intrusões, emitindo alerta imediato.
- **Conforto:** Acende automaticamente a luz quando o morador se levanta à noite.
- **Monitoramento:** Mostra a temperatura ambiente em tempo real.
- **Acessibilidade:** O som agudo e o alerta visual ajudam a ter uma visualização melhor da situação.