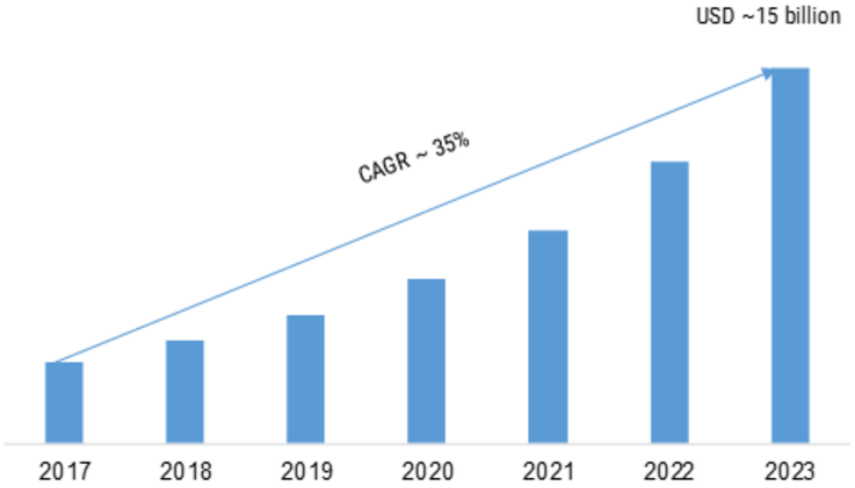
실시간으로 제어가능한 스마트 자동화 테스터기

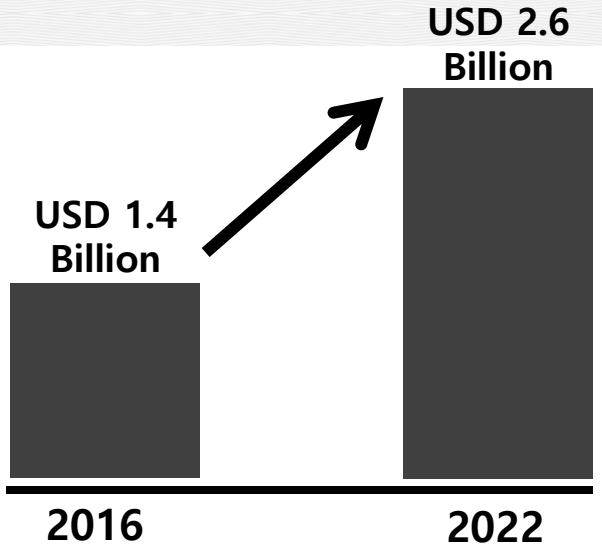
팀원

김유화
정우성
장혜주

01 작품의 배경



[글로벌 플렉서블 디스플레이 시장, 2017-2023(USD Billion)] [1]



[폴리이미드 필름 마켓, 2016-2022] [2]

- **플렉서블한 재료**: 최근, 많은 웨어러블 제품들과 디스플레이 분야에서 플렉서블한 재료(폴리이미드, 폴리프로필렌 등)가 주목을 받고, 이를 연구하기 위한 테스트 장치의 수요가 증가하는 추세
- **멀티 테스터**: 또한, 기존의 테스트 장치는 여러가지 테스트를 한 기계로 수행할 수 없어, 각각의 테스트를 수행하기 위해 높은 비용의 실험 장치를 테스트 목적에 맞도록 각각 구매해야 하는 문제점
- **자동화 공정**: 수 천,수 만 번의 테스트 과정에 자동화를 이용하여 생산성을 극대화 하기위해, 즉 능률적으로 진행하기 위해

02 작품의 목적

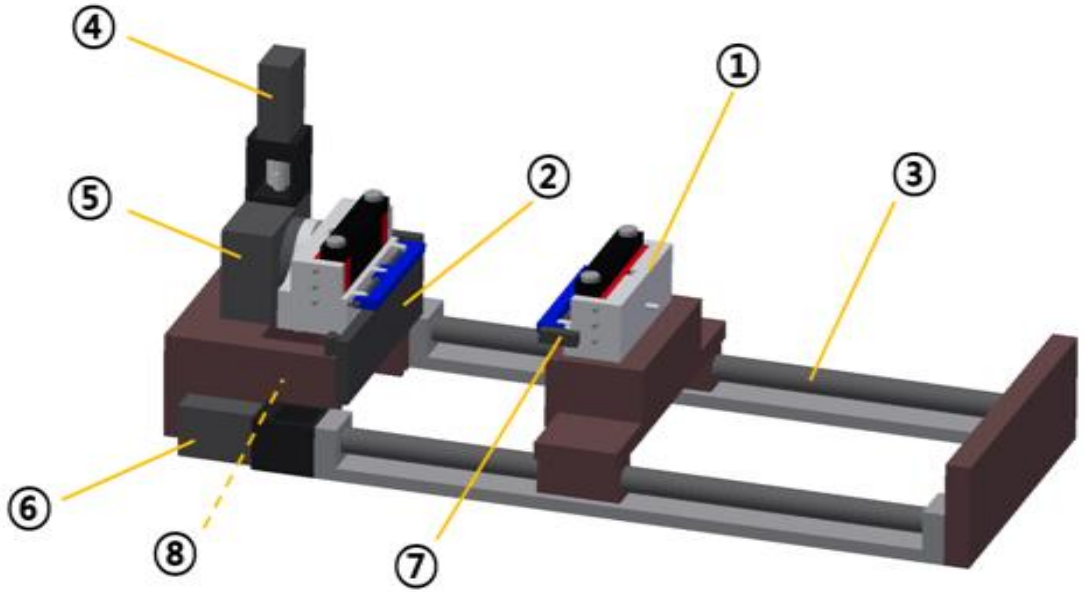
기존 문제점

- 테스트 기계를 각각 구매해, 높은 비용 발생
- 사람의 수작업의 경우, 해당 작업자의 숙련도나 건강, 집중 상태 등의 인적요인에 의해 제품의 품질에 어느 정도의 편차 발생
- 테스터 장비에 맞는 제품만 검사 가능한 문제점

우리의 목표

- **멀티 테스트** : 밴딩, 폴딩, 인장, 비틀림 테스트 등을 하나의 테스트장치에서 수행
- **연결성** : ICT기술을 이용해 테스트(품질 검사)까지의 과정을 연결 및 상호 소통
- **유연성** : 다양한 시편의 크기 및 재질을 대상으로 테스터가 가능한 유연한 체계
- **지능성** : 정보(테스트 결과값) 수집 및 축적을 통해 실시간 분석 및 제어

03 해결방법
멀티 테스트 기능



<전체 제품 그림>

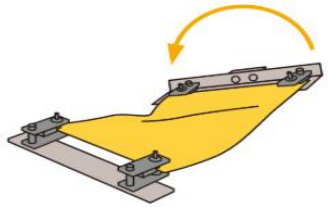
1. **전체 사이즈**: 625mm x 270mm x 260 mm
2. **구성**

   ① 스테이지 ② 고정 수단 ③ 볼 스크류 ④ 비틀림 모터

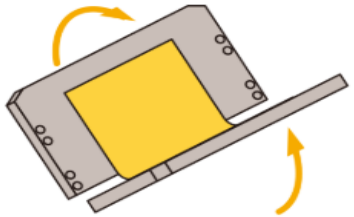
   ⑤ 기어 박스 ⑥ 인장 모터 ⑦ 각도 조절 모터

   ⑧ 모터 고정판

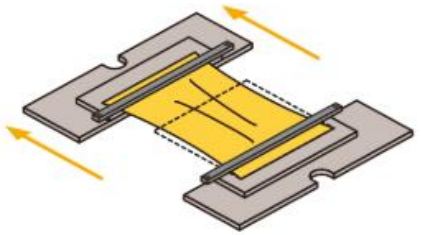
비틀림 테스트



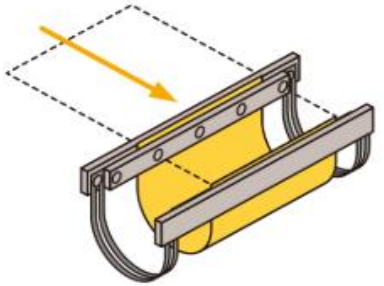
접힘 테스트



인장 테스트

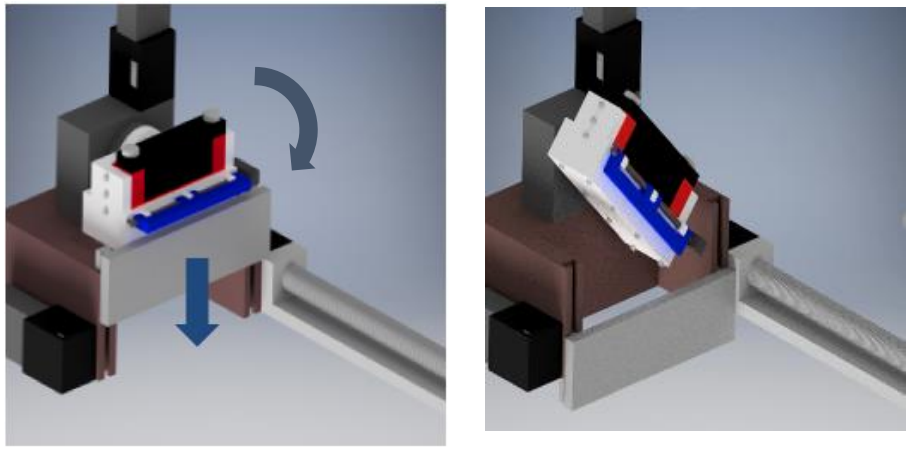


밴딩 테스트

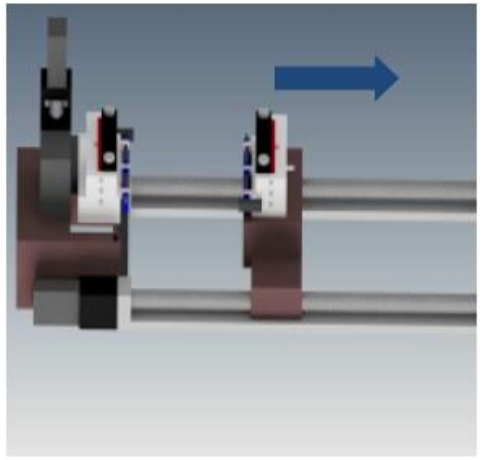


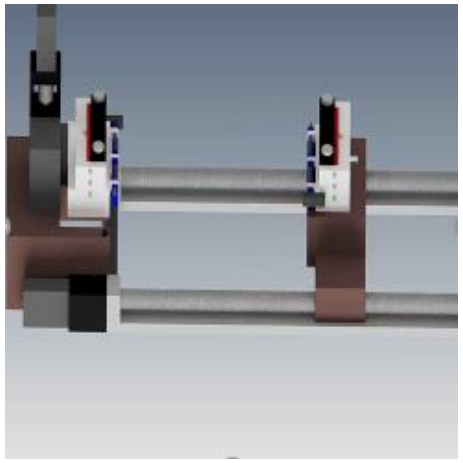
03 해결방법
멀티 테스트 기능

비틀림 테스트

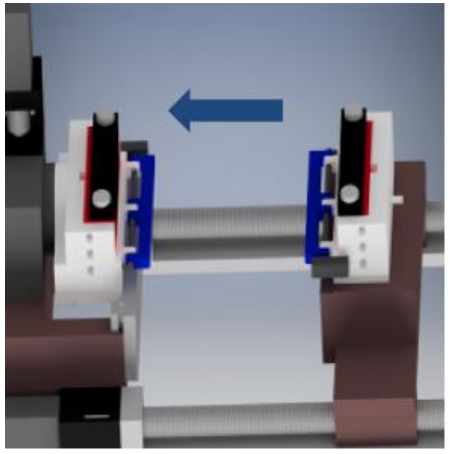


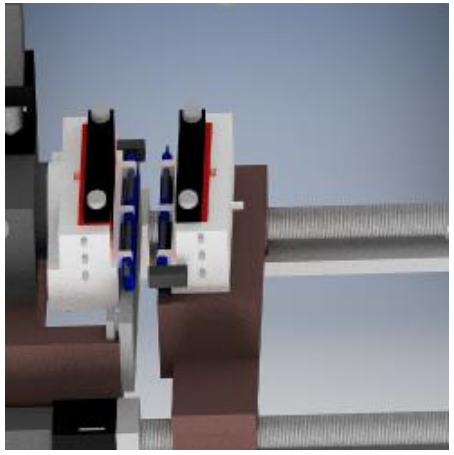
인장 테스트



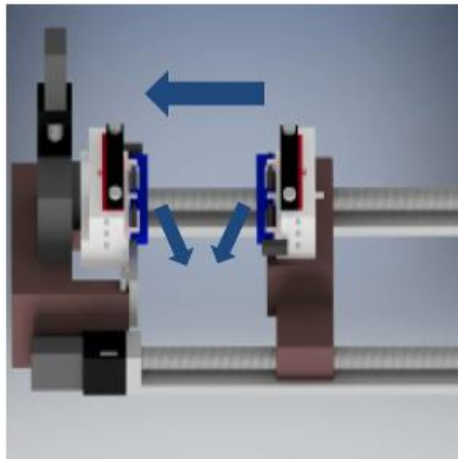


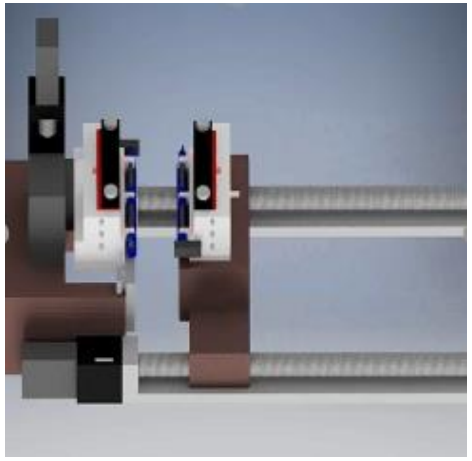
접힘 테스트





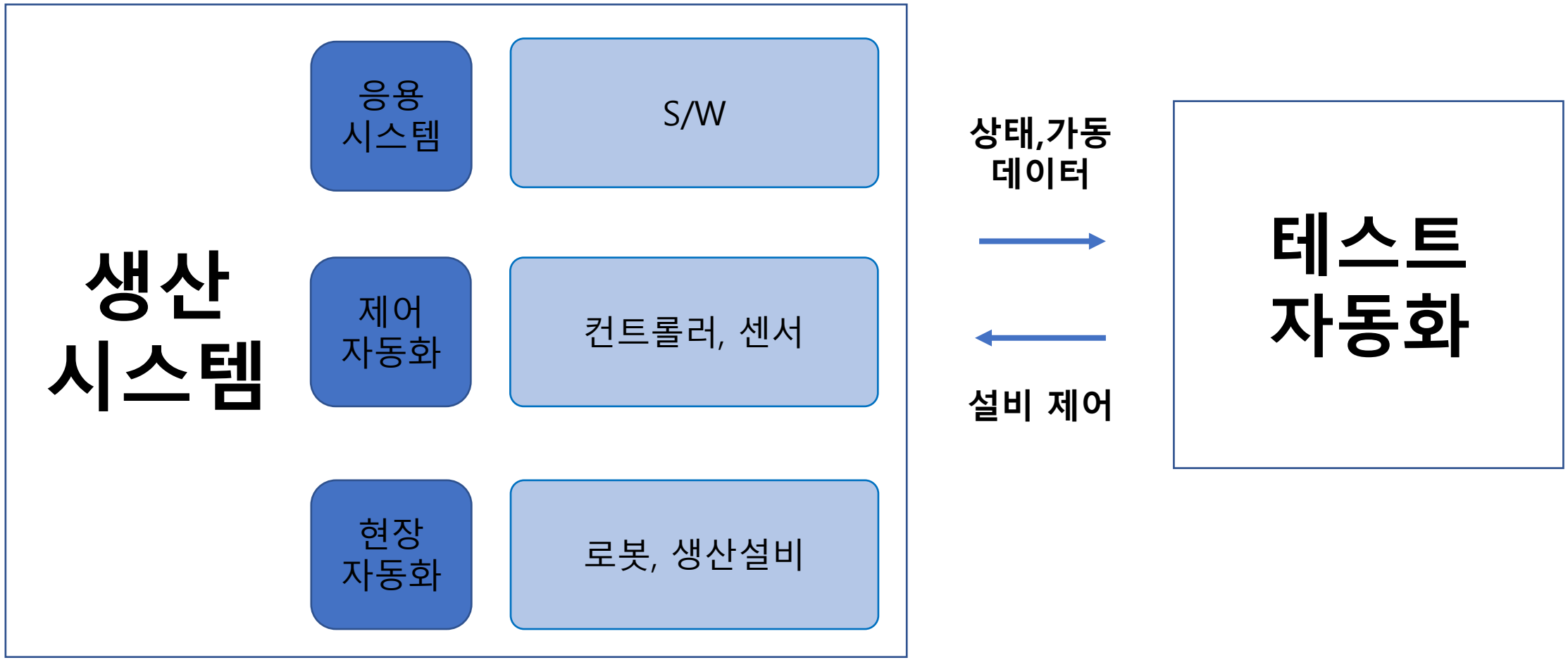
밴딩 테스트





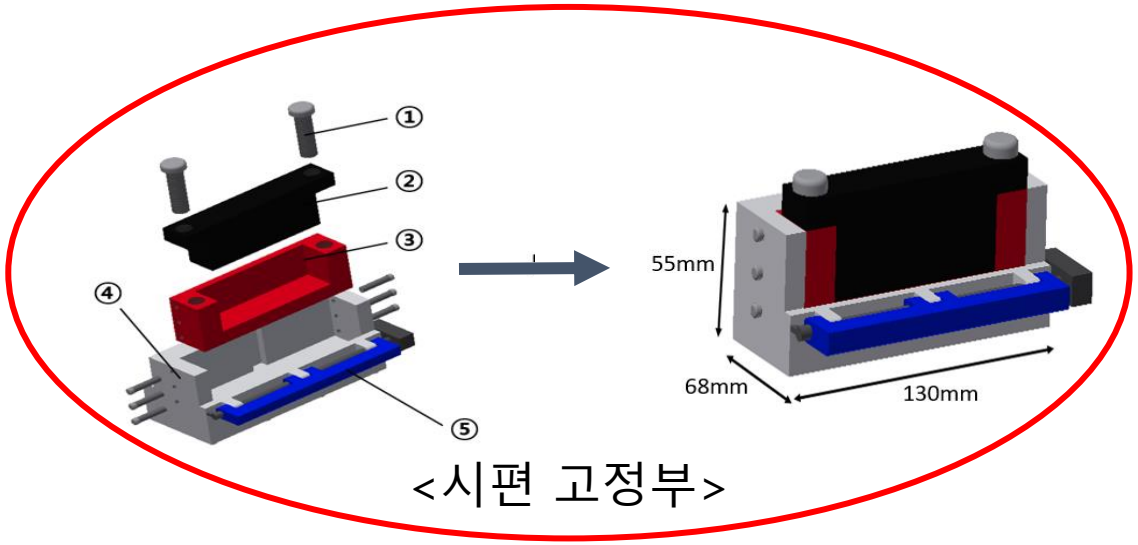
03 해결방법
연결성

- 연결성 : ICT기술을 이용해 테스트(품질 검사)까지의 과정을 연결 및 상호 소통



03 해결방법
유연성

- 유연성 : 다양한 시편의 크기 및 재질을 대상으로 테스터가 가능한 유연한 체계



<시편 고정부>

1. **전체 사이즈**: 55mm x 68mm x 55mm
2. **구성**
   : ① 나사 ② 상부 스테이지 ③ 하부 스테이지
   ④ 홀더 ⑤ 각도 조절 판
3. **재료**: ABS 수지(몸체 & 탈 부착 보드)/EPDM 고무(접촉면)
4. **홀딩방식**: 오버 클램프 & 테이프

- 나사를 이용한 조임 방식으로 제품의 **두께** 허용 범위가 넓습니다.
- 제품의 **길이** 또한 130mm 이내면 모두 확실하게 고정 가능합니다.

➔ 현재 목표로 하고 있는 웨어러블과 디스플레이 분야 외 에도 휴대폰 배터리 같은 작은 사이즈의 시편까지 테스트를 할 수 있습니다.

03 해결방법 지능성

- 지능성 : 정보(테스트 결과값) 수집 및 축적을 통해 실시간 분석 및 제어

IoT 디바이스 (테스터기)

온/습도
시편길이
테스트 속도
횟수

LTE
WiFi
Bluetooth
Zigbee
Ethernet

IoT 플랫폼

Data 수집/전달	Data 저장	Data 활용
MQTT CoAP 산업표준 http://	MariaDB redis mongoDB	
• 통신 프로토콜 지원 • Device 인증	• Device/Data 관리 • 대용량 Data 처리	• 센싱데이터 모니터링 • AI·빅데이터 분석 • 다음 테스터시 활용

04 기대효과

1 하나의 장치만 사용하여 공간 효율성 & 가격 경쟁력 확보

- 벤딩, 폴딩, 인장, 비틀림 테스트 등을 하나의 테스트 장치에서 수행할 수 있도록 하되, 공간의 낭비와, 간편화된 구성을 가졌습니다.
- 다수의 테스트 기계를 구매할 필요가 없어 합리적인 가격이 예상됩니다.

2 시편의 크기 허용성이 높아 다양한 산업분야에 진출가능

- 현재 목표로 하고 있는 웨어러블과 디스플레이 분야 외 에도 휴대폰 배터리 같은 작은 사이즈의 시편까지 테스트를 할 수 있습니다.

04 기대효과

3 동시 테스트 진행기술로 기술경쟁력 우위

- 테스트를 동시에 구현할 수 있는 테스트 방식(예: 스트레칭과 토션 테스트)을 이용하여 얻을 수 있는 테스트 결과 값이 더 많아집니다.
- 보통 한대의 장치로 하나의 테스트만 시행하기에 기술적으로 경쟁력이 있습니다.

4 시장의 안정적 수요확보

- 최근 코로나 19사태로 인한 마스크 수요의 폭발적 증가로, 마스크 또한 저희 장치로 제품을 테스트 할 수 있습니다.
- 최근 웨어러블 디바이스와 디스플레이 등의 제품이 플렉서블하게 출시되고 있습니다. (예: 갤럭시 폴드)

04 기대효과

5 테스트 도중에 실시간으로 시편의 손상 파악가능

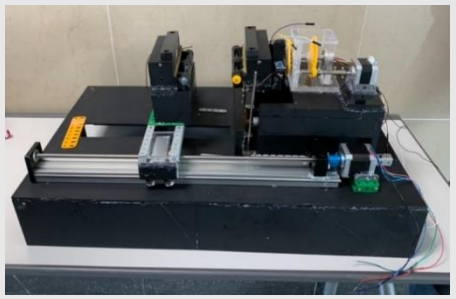
- 전도체인 시편의 경우 전기전도도를 실시간으로 측정함으로써 손상이 되었는지 파악할 수 있습니다.
- 또한, 테스트를 수행하는 도중에 시편을 가열함으로서 시편의 열에 의한 변형률 또한 측정할 수 있습니다.

6 IoT기술을 사용하여 효율적인 테스트가능

- 테스트 공정 자동화로 가동 효율성 증대
- 데이터 분석을 기반으로 운영현황 파악을 통한 효율적 관리

05 결론
Proto-type 제작

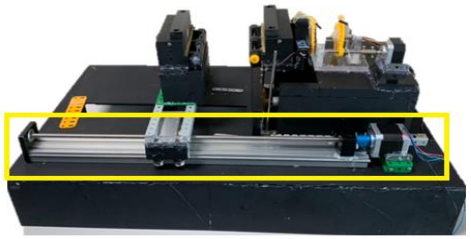
- **외부 구조 제작 (3D Print)**

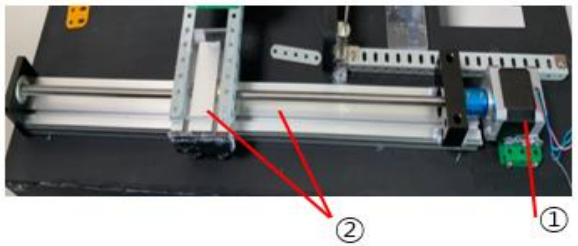


장비	Cubicon Single Plus
재료	PLA
출력 시간	총 36.9 시간
색상	Black

[3D printing specification]

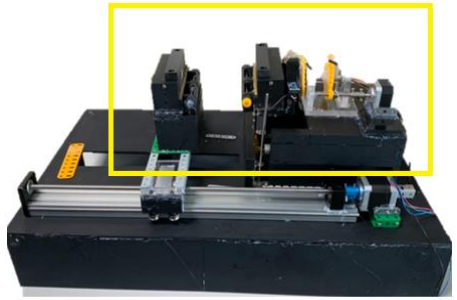
- **Stretching Test**

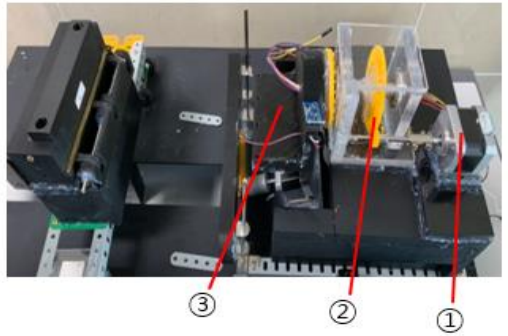




- 1번의 스텝 모터가 전원을 공급받으면 2번의 볼 스크루를 움직여 선형 운동을 구현해 Stretching Test를 시행

- **Torsion Test**



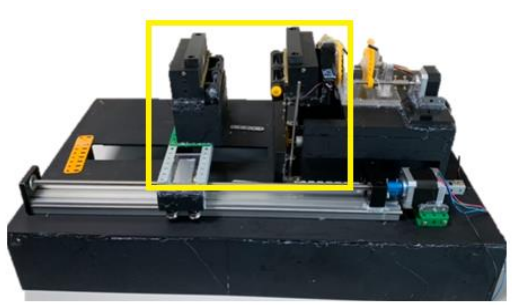


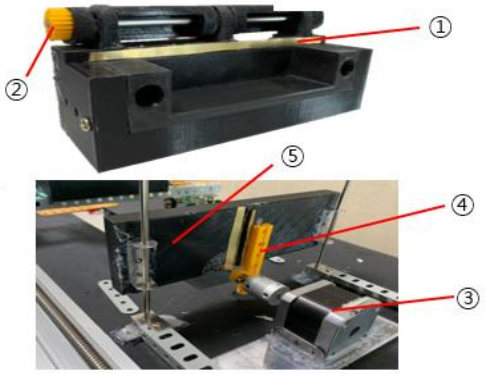
- 1번의 스텝 모터는 비틀림 시험의 각도를 조정하고 동력을 제공
- 기어박스 2는 아크릴 플레이트를 가공해 만든 것으로 모터 1에서 공급되는 출력을 효율적으로 전달

05 결론
Proto-type 제작

- **Bending Test**

(곡률반경 R 값을 조절 가능하게 하여 Folding Test 또한 구현)





- 1번의 금속판은 전기전도도를 측정
- 2번의 DC 서보 모터는 스테이지의 U자형 휨 각도(곡률반경 R 값)을 조절하기 위한 동력을 전달
- 4번은 지지판을 움직이기 위해 회전 운동에서 선형 운동으로 동력을 바꾸는 랙 기어

```
static void _main(void *arg) {
  while (1) {
    if ( Serial.available() ) {
      String s = Serial.readStringUntil('\n');
      short type = s.substring(0, 1).toInt();

      // set size
      if ( type == MODE_SET_SIZE ) {
        // sz is size of the length, as 'mm'
        sz = s.substring(1, s.indexOf(']')).toInt();
        linearMotor.moveTo(sz * 45);
        linearMotor.setSpeed(200);
        while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
          linearMotor.runSpeedToPosition();
        }
        linearMotor.setCurrentPosition(0);
```

<스텝 모터 코드>

```
// stretching
} else if ( type == MODE_STRETCHING ) {
    const int len = s.substring(1, s.indexOf(']')).toInt();
    const int duration = s.substring(s.indexOf('[') + 1, s.lastIndexOf(']')).toInt();
    const int numberOfTest = s.substring(s.lastIndexOf('[') + 1).toInt();

    writingType = 1;
    standard_millis = millis();
    for (int i=0; i<numberOfTest; i++) {
      const long motorPosition = linearMotor.currentPosition();
      const float spd = (float) len * 45 / (float) duration;
      linearMotor.moveTo(len * 45);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
      linearMotor.moveTo(motorPosition);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
    }
    writingType = 2;
```

<인장 테스트 코드>

```
// torsion
} else if ( type == MODE_TORSION ) {
    const int angle = s.substring(1, s.indexOf(']')).toInt();
    const int duration = s.substring(s.indexOf('[') + 1, s.lastIndexOf(']')).toInt();
    const int numberOfTest = s.substring(s.lastIndexOf('[') + 1).toInt();
    const float spd = (647.0 * (float) angle) / (252.0 * (float) duration);

    plateMotor.moveTo(273);
    plateMotor.setSpeed(200);
    while (plateMotor.currentPosition() != plateMotor.targetPosition()) {
      plateMotor.runSpeedToPosition();
    }
    vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);

    writingType = 1;
    standard_millis = millis();
    for (int i=0; i<numberOfTest; i++) {
      tortionMotor.moveTo((long) (647.0 * (float) angle / 252.0));
      tortionMotor.setSpeed(spd);
      while (tortionMotor.currentPosition() != tortionMotor.targetPosition()) {
        tortionMotor.runSpeedToPosition();
        log_value = 252.0 * (float) tortionMotor.currentPosition() / 647.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
      tortionMotor.moveTo(0);
      tortionMotor.setSpeed(spd);
      while (tortionMotor.currentPosition() != tortionMotor.targetPosition()) {
        tortionMotor.runSpeedToPosition();
        log_value = 252.0 * (float) tortionMotor.currentPosition() / 647.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
    }
    writingType = 2;

    plateMotor.moveTo(0);
    plateMotor.setSpeed(200);
    while (plateMotor.currentPosition() != plateMotor.targetPosition()) {
```

<비틀림 테스트 코드>

```
// u-shape bending
} else if ( type == MODE_U_SHAPE_BENDING ) {
    const int len = s.substring(1, s.indexOf(']')).toInt();
    const int duration = s.substring(s.indexOf('[') + 1, s.lastIndexOf(']')).toInt();
    const int numberOfTest = s.substring(s.lastIndexOf('[') + 1).toInt();
    const float spd = (float) (sz - len) * 45 / (float) duration;

    writingType = 1;
    standard_millis = millis();

    servo_msec = (long) ((float) duration * 11.11);
    servo_running = true;

    for (int i=0; i<numberOfTest; i++) {
      const long motorPosition = linearMotor.currentPosition();
      linearMotor.moveTo(sz - len) * 45);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
      linearMotor.moveTo(motorPosition);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
    }
    servo_running = false;
    writingType = 2;
```

<U자형 휨 테스트 코드>

```
// folding
} else if ( type == MODE_FOLDING ) {
    const int duration = s.substring(1, s.indexOf(']')).toInt();
    const int numberOfTest = s.substring(s.indexOf('[') + 1).toInt();
    const float spd = (float) sz * 45 / (float) duration;

    writingType = 1;
    standard_millis = millis();

    servo_msec = (long) ((float) duration * 11.11);
    servo_running = true;

    for (int i=0; i<numberOfTest; i++) {
      const long motorPosition = linearMotor.currentPosition();
      linearMotor.moveTo(sz * 45);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
      linearMotor.moveTo(motorPosition);
      linearMotor.setSpeed(spd);
      while (linearMotor.currentPosition() != linearMotor.targetPosition()) {
        linearMotor.runSpeedToPosition();
        log_value = (float) linearMotor.currentPosition() / 45.0;
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
    }
    servo_running = false;
    writingType = 2;
```

<접힘 테스트 코드>

```
static void servoMotor(void *arg) {
  while (1) {
    if ( servo_running ) {
      for (int i=SERVO_MIN; i<=SERVO_HALF; i++) {
        pwm.setPWM(M1, 0, i);
        pwm.setPWM(M2, 0, i);
        vTaskDelay((servo_msec * configTICK_RATE_HZ) / 1000L);
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
      for (int i=SERVO_HALF; i>=SERVO_MIN; i--) {
        pwm.setPWM(M1, 0, i);
        pwm.setPWM(M2, 0, i);
        vTaskDelay((servo_msec * configTICK_RATE_HZ) / 1000L);
      }
      vTaskDelay((500L * configTICK_RATE_HZ) / 1000L);
    }
  }
}
```

<서보 모터 코드>

06 역할분담

정우성

구동 시스템 엔지니어

- 모터 타입 조사
- 밴딩 구동 코딩
- 폴딩 구동 코딩
- 인장 구동 코딩
- 비틀림 구동 코딩
- 스테이지 디자인 엔지니어와 협력하여 모터와 회로 디자인 설계

장혜주

스테이지 디자인 엔지니어

- 스테이지 디자인
- 시편 고정방법 고안
- 3D CAD 모델링
- 3D Print를 이용한 부품 출력
- 구동 시스템 엔지니어와 협력하여 전체 부품 배치 및 각 테스트 기능 시험

정우성

시장 분석 및 총괄 매니저

- 시장 조사 – 기존 멀티 테스터 분석
- 고객 요구사항 분석 후 중요도 파악
- 타겟 샘플 조사
- 구동 시스템 엔지니어, 스테이지 엔지니어와 협력하여 테스트 기능 설계

07 참고 문헌

[1] https://www.marketresearchfuture.com/reports/flexible-display-technology-market-2302

[2] https://www.mordorintelligence.com/industry-reports/polyimide-films-market