

- **DECELERATOR AND GLIDING SYSTEMS**
- **CUSTOMIZED GUIDANCE, NAVIGATION & CONTROL SOLUTIONS**



**GUIDED RECOVERY AT A FRACTION
OF THE WEIGHT AND VOLUME
OF TRADITIONAL SYSTEMS**



Decelerator and Gliding Systems

Guided Gliding Systems

MMIST's new custom parafoil design is perfectly suited to vehicle recovery applications and offers some significant advantages over standard ram-air parafoils in the guided parachute recovery market.

Advantages include:

- Lower weight
- Lower pack volume
- Highly variable glide ratio (for precision purposes)

Custom Aerodynamic Decelerator Systems

For applications requiring multiple stage systems, MMIST has supported customer re-entry profiles with the design and manufacture of high altitude aerodynamic decelerator systems specific to their application.

Customized Guidance, Navigation and Control Solutions

Based on MMIST's proven Sherpa Precision Aerial Delivery System:

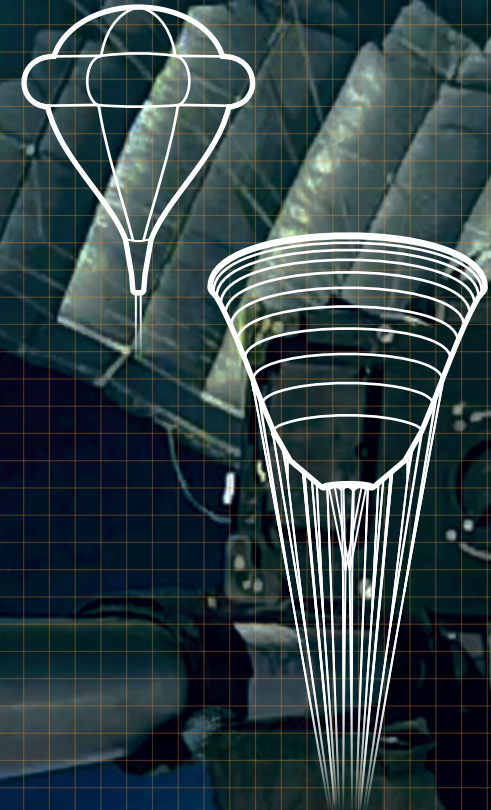
- Autonomous GPS Guidance augmented with Accelerometer & Gyro Sensors
- HALO/HAHO (opening altitude determined by GPS Altitude, Pressure Altitude, or Time Delay)
- Features waypoints & 'No Fly' Zones
- Multi Constellation GPS Support: NAVSTAR, GALILEO, GLONASS

Easy Integration into Third Party Architectures / Systems

Access Sherpa's core circuit board components and proprietary guidance software available as OEM component board for seamless integration into existing third party architectures.

Key components include:

- Basic Navigation Board (Parachute Control Board / Core CPU / Guidance Firmware) and (optional / as required):
- Sensor Expansion Board (GPS, Accelerometer, Gyro Sensor Suite)
- Motor Controller Board
- Parachute Control Actuators
- Custom Bus Interface
- Ruggedized Packaging



```
var guidance;
function init guidance() {
  map = new maps.Map(document.getElementById('map'), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 6
  });
  infoWindow = new google.maps.InfoWindow({
  });
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(position) {
      var pos = {
        lat: position.coords.latitude,
        lng: position.coords.longitude
      };
      infoWindow.setPosition(pos);
      infoWindow.setContent('Location: ' + pos);
      infoWindow.open(map);
      map.setCenter(pos);
    }, function() {
      handleLocationError(positionError, infoWindow);
    });
  } else {
    handleLocationError(browserNotSupportedMessage, infoWindow);
  }
}

function handleLocationError(browserNotSupportedMessage, infoWindow) {
  infoWindow.setContent(browserNotSupportedMessage);
  infoWindow.setPosition({lat: 38, lng: -120});
  infoWindow.open(map);
}
```

